LIPIEC 198[illegible] CENA [illegible] ZŁ ISSN 0860-2614

KOMPUTER

4

POPULARNY MIESIĘCZNIK INFORMATYCZNY



S. SZCZYPKA

KOMPUTER NR 4

Popularny Miesięcznik Informatyczny – pismo miłośników i użytkowników mikrokomputerów redagują:

Marek Młynarski (red. nacz.)
Władysław Majewski (z-ca red. nacz.)
Grzegorz Eider (sekr. red.)
Elżbieta Bobrowska (z-ca sekr. red.)
Grzegorz Czapkiewicz (programy)
Stanisław Królak (dz. zagraniczny)
Jerzy Pusiak (kasety)
Zenon Rudak (sprzęt)
Darosław J. Toruń (gry)
Tomasz Zieliński (listy)
Krzysztof Krupa
oraz współpracownicy:
Andrzej Bączyński (Łódź), Rafał Brzeski, Marek Car, Andrzej Kadlof, Jarosław Kania, Agnieszka i Zbigniew Kasprzyccy, Krzysztof Kuryłowicz (Łódź), Jacek A. Likowski, Juliusz Rawicz, Leszek Rudak, Grzegorz Szewczyk, Jakub Tatarkiewicz, Piotr Norbert Tymochowicz, Roland Wacławek (Katowice), Tadeusz Wilczek, Andrzej Załuski (Kraków)

Redakcja graficzno-techniczna:
Stefan Szczypka (kier.)
Małgorzata Luzińska
Beata Maruszewska

korekta: Maria Omiecińska, Romualda Miarecka

Wydawca: Krajowe Wydawnictwo Czasopism RSW "Prasa-Książka-Ruch", ul. Noakowskiego 14, 00-666 Warszawa, tel. centr. 25-72-91 do 93
Redakcja: ul. Mokotowska 48, 00-543 Warszawa, tel. 21-76-58 telex 815664 cestud pl (gości nas Warszawskie Centrum Studenckiego Ruchu Naukowego ZSP)
Skład i druk: Prasowe Zakłady Graficzne, Łódź, ul. Armii Czerwonej 28. Zam. 1637/86
Cena: 100 zł, P-62

Prenumerata: kwartalnie – 300 zł, półrocznie – 600 zł, rocznie – 1200 zł. Prenumeratę od instytucji przyjmują oddziały RSW, a od osób prywatnych poczta (na wsi także doręczyciele). Prenumeratę ze zleceniem wysyłki za granicę (droższą o 50% dla osób prywatnych i o 100% dla instytucji) przyjmuje Centrala Kolportażu RSW, ul. Towarowa 28, 00-958 Warszawa, NBP XV O/M W-wa 1153-201045-139-11.
Prenumerata przyjmowana jest na II, III i IV kwartał oraz na II półrocze z miesięcznym wyprzedzeniem, a na rok następny do 10 listopada.

Ogłoszenia przyjmuje Biuro Reklamy, ul. Mokotowska 5, tel. 25-35-36; adres dla korespondencji w sprawach ogłoszeń: Noakowskiego 14, 00-666 Warszawa. Zamawiając ogłoszenia listownie należy podać datę i miejsce wpłaty (konto KWCz: NBP III O/M W-Wa 1036-5294 z zaznaczeniem "ogłoszenie w KOMPUTERZE").

1 cm² ogłoszenia kosztuje 300 zł, najmniejsze ogłoszenie – 2100 zł, cała strona – 200 tys. zł; kolor dodatkowy – 30% drożej, pełna gama barw – 100% drożej. Za treść ogłoszeń redakcja nie odpowiada.

Nakład 150.000 egz.
Nr indeksu 36-345 ISSN 0860-2514

CIŚNIENIE

Zwarty tłum, karnie i cierpliwie czekający w wielometrowej kolejce, która po kilku zakrętasach prowadzi przez ciasne schody (też z jednym zakrętem) do WEJŚCIA. Za grzecznymi, choć nieustępliwymi strażnikami jest SALA – w niej tu i ówdzie migocą monitory, brzmi "elektroniczna" muzyczka, wszędzie kłębią się gromady zainteresowanych MIŁOŚNIKÓW, nie wiadomo, czy przywiedzionych tu ciekawością, modą, chęcią wiedzy? Są oni gotowi grzecznie czekać kwadranse, przechodzące w godziny na wejście, żeby... No właśnie, czego się spodziewają? Że w tym tłoku cokolwiek zobaczą? Że będą mogli własnoręcznie ponaciskać kilka klawiszy? Że uda się nawiązać kontakty z bratnią duszą, też posiadaczem komputera, który ma być może to, czego ja nie mam? Takie i inne przypuszczenia można snuć dość długo, a zarysowany powyżej obrazek z pokazu jakiegokolwiek sprzętu komputerowego powtarza się od długiego już czasu. No więc dobrze to czy źle; czy to olbrzymie zainteresowanie powinno nas cieszyć?

Ten, komu udało się wejść do SALI, natychmiast może się zorientować, że szanse dorwania się do klawiatury są znikome, ale jeżeli jest się odpowiednio wyrośniętym (pożądane ponad 180 cm), ponad głowami innych można dostrzec niepozorną skrzynkę z klawiaturą, a na ekranie monitora różne różności, które może ona zrobić.

Ponieważ FANATYK zna na pamięć nie tylko podstawowe, ale i produkcyjno-techniczne dane, z podziwem obserwuje i monitor, i skrzynkę komputera. W wyobraźni widzi siebie, siedzącego późną nocą w kuchni lub łazience (żeby nie budzić innych domowników) i z należytą ostrożnością wkładającego dyskietkę w odpowiednią szczelinę, przekręcającego dźwignię i podziwiającego właśnie to, co sam zaprojektował, zaprogramował i uruchomił.

Jako żywo przypomina to lizanie cukierka przez szybę znanego z różnych opowieści o tych dawnych czasach, kiedy to pragnienia dzieci i dorosłych były realne jedynie w sferze marzeń. Dziś pożądanie nie dotyczy co prawda kawałka ekstraktu z buraka cukrowego, a kilku kawałków odpowiednio ułożonego krzemu, ale ciągle jeszcze jest dla większości realne tylko w marzeniach.

I dlatego widok grubaśnej i wijącej się kolejki czekających na wejście na pokaz mikrokomputerów wcale mnie nie cieszy. Nie wierzę bowiem, że tak cierpliwie stoją w kolejce tylko posiadacze sprzętu, przypuszczam, że proporcje "staczy" kształtują się następująco: 1/4 powodowana jest panującą modą na mikrokomputery, 1/2 przychodzi, żeby liznąć tego sprzętu przez szybę, a jedynie pozostała 1/4 ma w planach wymianę doświadczeń. Te proporcje są zresztą całkowicie umowne i mogą się kształtować zupełnie ina-

czej, chociaż jestem przekonany, że moda nie stanowi przyczyny zachowań więcej niż 1/4 widzianych w kolejce.

Problem jest jednak poważny i wiążą się z nim istotne implikacje socjologiczno-społeczne. Jak pamiętam z najnowszej historii, podobne namiętności wzbudzała jedynie sprawa popularnego, ogólnie dostępnego samochodu. Każdy wie dobrze, jak się ona skończyła. Dziś stawka jest znacznie niższa, a nadzieja, którą się żywię, oparta jest na przekonaniu, że nauczeni i cały czas bici po kieszeni zaciągniętymi długami (a część dolarowej forsy miała wspierać naszą tanią i dostępną motoryzację), nie pójdziemy już w kierunku zaciągania nowych, wysoko oprocentowanych i płatnych w walucie pożyczek. A jeżeli musimy już handlować z tymi, którzy nie chcą naszych, przecież z nazwy pochodzących od złota, pieniędzy, to wszak istnieją inne sposoby handlowania, może nieco historycznie bardziej prymitywne – towar za towar. Popierajmy takie transakcje, pod warunkiem, że nie zubożą naszego rynku. Ciągle jednak powtarza się pytanie: co począć z obecnym fenomenem zainteresowania informatyką?

KONIECZNE SĄ SZYBKIE I SKUTECZNE DECYZJE W SFERZE PRODUKCJI !!!

Wiemy, że Polska jest opóźniona w stosunku do przodujących państw świata. Specjaliści, m.in. na naszych łamach, twierdzą jednak, że dystans ten jest do odrobienia. Co prawda podany w tych wypowiedziach czas potrzebny na doścignięcie czołówki jest mocno dyskusyjny, ale zdecydowanie uważam, że można i trzeba to robić, a im szybciej, tym lepiej. O pewnych projektach pisałem w poprzednim numerze "Komputera", niestety ciągle jeszcze nie mogę podać szczegółów tej transakcji. Nie ulega jednak wątpliwości, że nie rozwiążemy sprawy produkcją kilkuset czy kilku tysięcy mikrokomputerów rocznie. Całym sercem popieram każdy kontrakt, który zapewni na naszym rynku dziesiątki lub setki tysięcy komputerów po przystępnych cenach, na początek rzędu 80-60 tys. zł za mikrokomputer o możliwościach przewyższających, a jednocześnie zgodnych z ZX Spectrum. Tylko takich rozmiarów produkcja może rozładować owo ciśnienie, osiągające obecnie wartości maksymalne: 1 mikrokomputer/ 1 zainteresowanego/ 1 rok.

Z przyjemnością mogę stwierdzić, że gdy piszę te słowa, nieźle jest w sferze informacji – nadbudowy komputeryzacji. Jak ogólnie wiadomo, zespół tworzący tzw. starego Bajtka podzielił się i w oparciu o sukcesy wydawanego w 1985 roku przez KWCz "Bajtka" powstał "Komputer" (jedyne samodzielne pismo informatyczne) i "Bajtek", już jako dodatek do "Sztandaru Młodych". Z racji wspólnych korzeni i "Komputer", i "Bajtek" należą do pierwszych regularnie wydawanych pism informatycznych. Wydawnictwo "Sigma", które na łamach "Informatyki" prowadziło "Mikroklan" (zdecydowanie najstarszy dodatek informatyczny, redagowany przez Andrzeja J. Piotrowskiego) prawdopodobnie będzie wydawało zeszyty "Mikroklanu". Do wydawnictw tych dołączył "IKS" – dodatek do "Żołnierza Wolności", a następnie "Konkret" – cykl zeszytów wydawanych przez Stronnictwo Demokratyczne. Rubryki tematyczne, często o charakterze osobnych części pisma, zamieszczają m.in. "Problemy", "Wiedza i Życie", Młody Technik", "Przegląd Techniczny", "Horyzonty Techniki" i wiele innych pism. Zapowiadane są nowe rubryki i dodatki.

Wszystko razem tworzy w miarę optymistyczny obraz. Jednak wiadomości, które przekazujemy w "Komputerze" i które są w pozostałych dodatkach i rubrykach informatycznych, muszą mieć wsparcie w ogólnodostępnym, tanim sprzęcie. Jest to podstawowy warunek, żeby przekazywane treści nie były traktowane jak rozmowa gęsi z prosięciem. I stąd niestety ciągle aktualne wezwanie do przemysłu, handlu, decydentów, finansistów, spółek polskich, mieszanych i zagranicznych:

Udostępnijmy tani sprzęt w takich ilościach, żeby każdy miłośnik komputerów mógł go kupić, a wówczas będziemy mogli stwierdzić, że pierwszy krok w kierunku nowoczesności jest za nami!

MAREK MŁYNARSKI



MENU !K

PRZEMYSŁ KOMPUTEROWY

(Rozmowa z dr. inż. JERZYM DYCZKOWSKIM, dyrektorem Zespołu Elektroniki w Urzędzie Postępu Naukowo-Technicznego i Wdrożeń).

– Jak będzie wyglądał krajowy rynek mikrokomputerowy w 1990 roku?

– Znając działania podjęte przez zakłady przemysłu komputerowego oraz Urząd Postępu Naukowo-Technicznego i Wdrożeń mogę mówić o ofertach na zamówienia rządowe, które zostały przyjęte do dalszych negocjacji. W 1990 r. jest możliwe opanowanie produkcji mikrokomputerów profesjonalnych na poziomie 30-40 tys. sztuk rocznie oraz mikrokomputerów edukacyjnych w ilości 100-200 tys. sztuk rocznie. Konkretne ilości będą znane po negocjacjach i podpisaniu umów na realizację zamówień rządowych. Wstępnie przyjęto ofertę Zakładów Elektronicznych ELWRO przewidującą roczną produkcję 30 tys. mikrokomputerów ELWRO 800, 120 systemów TELEJS, zakładów POLKOLOR na osiągnięcie zdolności produkcyjnej 250 tys. precyzyjnych lamp monitorowych, zakładów ELZAB na 30 tys. sztuk monitorów ekranowych dla SM EMC, rodziny grafploterów z MERA STER, 50 tys. jednostek pamięci na dyskach elastycznych typu Slim-line z KFAP, tej samej ilości drukarek D-100M z MERA-BŁONIE. Nie są to wszystkie przyjęte oferty. Wiele innych zgłoszeń, bardzo ciekawych technicznie i dobrze przygotowanych ekonomicznie, jest obecnie analizowanych.

Nie potrafię jednak zdecydowanie scharakteryzować rynku mikrokomputerowego w 1990 roku. Bardzo dużo będzie zależeć od sytuacji płatniczej kraju i stabilności systemów wspomagających rozwój elektroniki.

Istotne znaczenie ma inny tryb prowadzenia inwestycji. Nie można uzyskać znaczących efektów bez 2-3-krotnego przyspieszenia realizacji inwestycji. W odróżnieniu od innych przemysłów, 5 lat w elektronice to epoka.

Ważnym czynnikiem może być przełamanie barier we współpracy naukowo-technicznej krajów socjalistycznych. Podpisanie Kompleksowego Programu Współpracy Naukowo-Technicznej Krajów RWPG do 2000 roku jest istotnym krokiem.

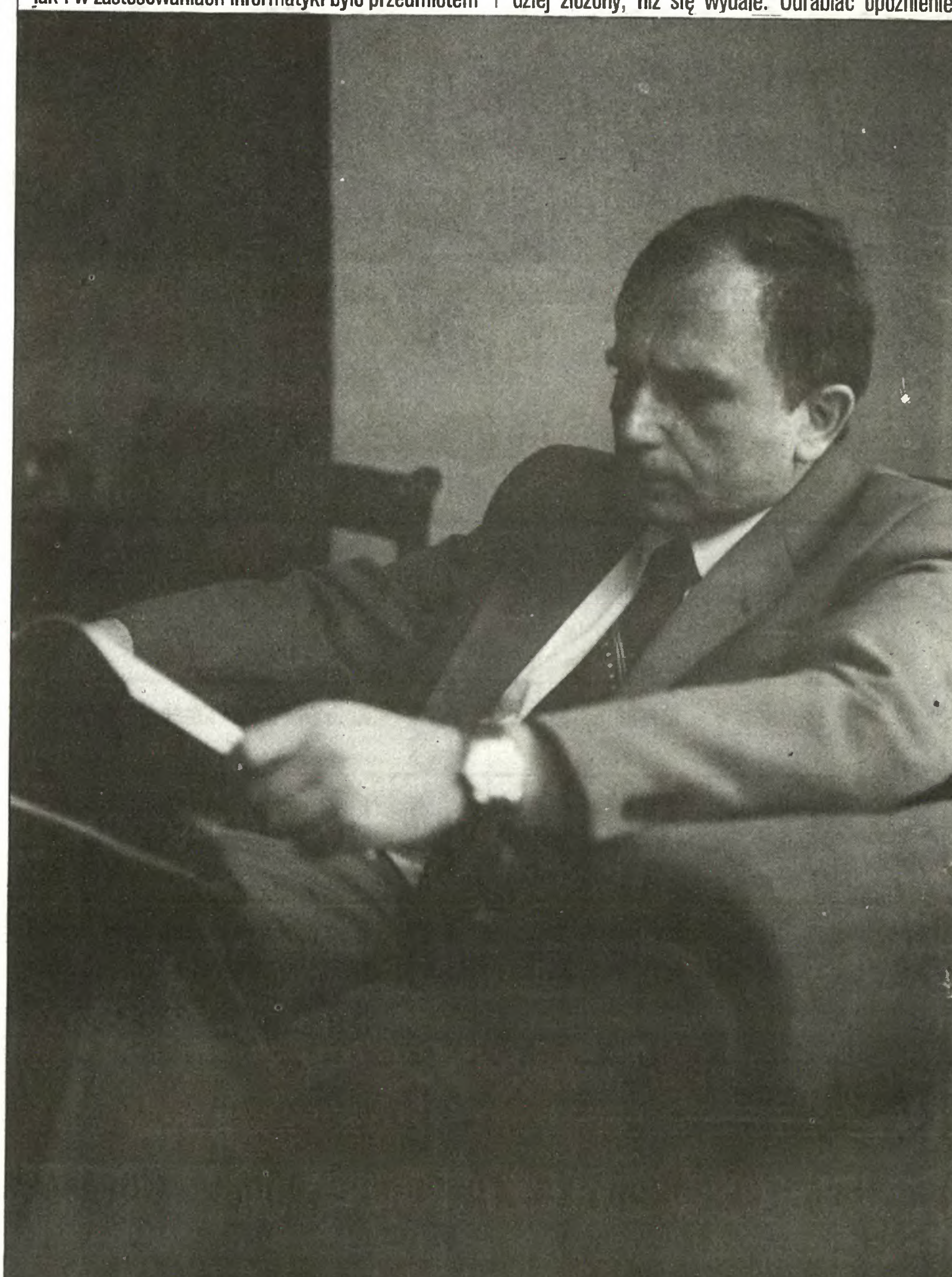
– Czy z tego wynika niechęć do współpracy naukowo-technicznej w elektronice z wysokorozwiniętymi krajami kapitalistycznymi?

– Nie, nie wynika. Każdą konkretną czy ogólną ofertę współpracy podejmujemy i próbujemy negocjować. Często mają one jedynie charakter rozeznaniowy i sondażowy. Bez trudności np. możemy kupić licencję na kremy do golenia najnowszej generacji, ale nie na grafoskop o parametrach... sprzed dziesięciu lat. Na największych zwolenników kupna licencji działa otrzeźwiająco zapoznanie się z istniejącymi zachodnimi przepisami dotyczącymi transferu technologii. Innymi słowy – chcielibyśmy współpracować z rozwiniętymi krajami kapitalistycznymi, ale nie mamy żadnych złudzeń. Zmiany przepisów embargowych są dokonywane wówczas, gdy kraje socjalistyczne opanowują daną technologię i utrzymywanie dotychczasowych ograniczeń jest niecelowe. Każda zmiana przepisów embargowych to sukces zaplecza technicznego, a nie "liberalizacja współpracy".

– Jak ocenia Pan opóźnienie Polski w stosunku do przodujących krajów świata? Czy w naszej sytuacji gospodarczej mamy szansę na jego nadrobienie?

– Opóźnienie Polski w sprzęcie komputerowym jak i w zastosowaniach informatyki było przedmiotem wielu analiz i nie sądzę, żebym mógł dodać coś istotnego. Dla wyrobów wprowadzanych do produkcji opóźnienie wyrażone w latach wynosi od 4 do 10 lat. Przy uwzględnieniu parametrów niezawodnościowych wyrobów opóźnienie jest jeszcze większe i jest uwarunkowane głównie przez bazę podzespołową, w tym układy scalone bardzo dużej skali integracji. Istotne znaczenie mają również niektóre technologie wytwarzania, narzędzia i maszyny specjalistyczne oraz pewne specyficzne technologie stosowane przy produkcji urządzeń zewnętrznych.

Problem zniwelowania opóźnienia jest o wiele bardziej złożony, niż się wydaje. Odrabiać opóźnienie,



to znaczy rozwijać się szybciej niż inni, nie obciążeni zadłużeniem i dysponujący doskonałą aparaturą, posiadający duże, doświadczone zespoły i mogący przeznaczyć np. 80 miliardów dolarów rocznie na rozwój informatyki. Zbyt często w Polsce wyrażana jest naiwna nadzieja, że wystarczy deklaracja i luka technologiczna zniknie jak za dotknięciem czarodziejskiej różdżki.

– *Wszak uchwalono i podpisano szereg ważnych dokumentów.*

– Ostatni rok przyniósł rzeczywiście szereg ważnych dokumentów, zwiększono także nakłady na prace badawcze. W 1985 r. rozwój nauki i postępu technicznego pochłonął 1,8% dochodu narodowego. W br. będzie to już 2,4%. Z tej kwoty ponad 20% przeznaczono na elektronikę, w tym także na komputery.

Bardzo istotne jest ujęcie zadań związanych z rozwojem przemysłu komputerowego i zastosowań informatyki w Programie PZPR oraz w ustaleniach X Zjazdu PZPR.

– *Czy możemy mówić o programie zastosowań informatyki?*

– To, o czym przed chwilą mówiłem, pomaga w przygotowaniu szczegółowych planów rozwoju produkcji sprzętu komputerowego i programu rozwoju informatyki. Taki program jest potrzebny przede wszystkim dla świadomego kierowania jej rozwojem w konkretnych miejscach gospodarki narodowej. Będzie on opracowany, choć jest to wyjątkowo trudne zadanie.

– *A jednak nie można powiedzieć, że w Polsce istnieje solidny dział gospodarki produkujący komputery i urządzenia zewnętrzne do nich...*

– W Polsce istnieje liczący się przemysł komputerowy niezależnie od tego, czy niektórzy publicyści zauważają to, czy nie. W 1984 r. wyprodukowano m.in. 177 minikomputerów SM4A, 427 mikrokomputerów MERA 60, 82 systemy przygotowania danych MERA 9150, 1765 pamięci na dyskach elastycznych, 9633 drukarek znakowych i 1268 drukarek wierszowych.

W 1984 r. wyeksportowano m.in. 113 minikomputerów SM4A, 351 mikrokomputerów MERA 60, 71 teleprocesorów przetwarzania danych, 300 pamięci taśmowych PT 305, 1186 drukarek wierszowych.

Natomiast w 1985 r. nastąpił wzrost produkcji tego przemysłu o 27,9%, licząc w cenach stałych. Nie sądzę, żeby przyrosty produkcji w kolejnych latach bieżącej pięciolatki były mniejsze. Produkcja sprzętu komputerowego będzie podjęta również przez nowych producentów. Znaczne, a nawet skokowe wzrosty nastąpią we wszystkich przedsiębiorstwach przemysłu komputerowego.

– *Eksport sprzętu komputerowego był wielokrotnie przedmiotem zdecydowanej krytyki?*

– Przedstawiciele przemysłu komputerowego niejednokrotnie wyjaśniali, że zagwarantowana umowami specjalizowanymi produkcja pozwala na wprowadzenie technologii obniżających koszty wytwarzania, co jest ważne i dla użytkowników krajowych. Uzyskiwane ceny eksportowe, wyższe od cen krajowych, pozwalają na inwestycje odtworzeniowe oraz inwestycje umożliwiające znaczne zwiększenie produkcji. Eksport wyrobów komputerowych w zamian za

▶ 44

NAUCZYCIELE I KOMPUTERY

Piotrków Trybunalski. Koniec marca. W budynku Oddziału Doskonalenia Nauczycieli kolejna sesja Studium Informatyki, zorganizowana przez Instytut Kształcenia Nauczycieli w Warszawie. Grupa pedagogów z całego kraju zdobywa merytoryczne i metodyczne przygotowanie do przyjęcia mikrokomputerów w szkolne mury. Poznają najpopularniejsze języki programowania – Basic, Logo, Pascal, i podstawy metodyki nauczania informatyki w związku z wprowadzeniem takiego przedmiotu nauczania jako fakultatywnego do programu szkół średnich już w tym roku.

Zainteresowanie uczestnictwem w Studium było olbrzymie. Nie przerażają nauczycieli stosunkowo trudne egzaminy. Co prawda oprócz zapaleńców są i sceptycy, ale ci wzbudzają pozytywną refleksję: nowy, efemeryczny eksperyment dydaktyczny wyrosły na fali mody czy triumfalne wkroczenie nowoczesności do szkoły (wreszcie...). Okazuje się, że są placówki oświatowe, dla których informatyka nie jest nowością ani modą...

Franciszek Fredek, III LO z Wrocławia:

"Mamy z pewnością najbogatsze tradycje informatyczne wśród placówek oświatowych w Polsce. 22 lata temu zostały utworzone pierwsze klasy programowania, które wykształciły kilka roczników absolwentów. Wielu z nich podjęło prace w ośrodkach obliczeniowych. W ostatnich oddziałach kierunku matematycznego elementy informatyki prowadzone są od 15 lat, a ostatnio w ramach wychowania technicznego objęto tym przedmiotem wszystkie klasy trzecie. Ogromnie owocuje współpraca z Instytutem Informatyki Uniwersytetu Wrocławskiego. Łącznie 8 klas uczęszcza do instytutu na zajęcia prowadzone pod okiem fachowców na dobrym sprzęcie. Od 2 lat mamy własną pracownię wyposażoną w trzy Spectrum, jeden Commodore, cztery Meritum i drukarki. Pragniemy wkrótce zakupić 10 Amstradów ze stacją dysków. Uczniowie przejawiają mnóstwo inicjatyw, napisali sporo programów dydaktycznych stosowanych na lekcjach, wykonali pióro świetlne i klawiaturę do Spectrum. Jeden z nich zdobył I nagrodę w konkursie na program dydaktyczny, rozpisanym przez PTI we Wrocławiu".

Inni mają bardziej podstawowe problemy. Anna Łaś, LO Kuźnice-Zakopane przy sanatorium SZRODIM:

"Mamy środki na zakup dwóch komputerów, ale co z tego? Jedynym rozwiązaniem wydaje się skorzystanie z usług komisu wysyłkowego w Łodzi. Trwa to jednak kilka miesięcy od momentu złożenia zamówienia".

Szkoła kupując komputer w komisie przepłaca, nawet do 50% wartości wolnorynkowej, jest to swego rodzaju haracz za możliwość otrzymania rachunku. Zamiast dwóch komputerów mogłaby mieć trzy, ale na ten trzeci zarabiają pośrednicy handlowi, którym wolno kupować towar od osób prywatnych. Wygląda to od strony oświaty na szastanie pieniędzmi, ale jest zwykłym "biciem przymusowym". Są, jak w każdym przypadku, inne drogi, lecz nie wszyscy mogą nimi chodzić.

Alicja Bujacz, LO w Pabianicach, woj. łódzkie:

"Nie mamy jeszcze komputerów. Dyrekcja LO czyni starania, aby porozumieć się z przedsiębiorstwem POLAM, które posiada komputery, by tam mogły odbywać się zajęcia informatyki".

Anna Suszyło, Zespół Szkół w Leżajsku, woj. rzeszowskie:

"Weszliśmy w kontrakt z zakładami Przemysłu Owocowo-Warzywnego w Leżajsku. Dyrektor jest dla nas bardzo życzliwy i w tym roku ma nam ofiarować kilka mikrokomputerów. Informatycy u niego zatrudnieni zaofiarowali pomoc. Dla nas to bardzo dużo".

Jacek Leszewski z Zespołu Szkół Leśnych w Warcinie, woj. słupskie:

"Bazę mamy niewielką, zaledwie dwa Spectrum 48K zakupione w BOMIS-ie i z tego, co wiem, jesteśmy w Słupskiem potentatem w tej materii. Zajęcia z mikrokomputerami prowadzimy tylko po lekcjach, przy ogromnym zainteresowaniu młodzieży. Dwie planowe godziny zajęć przedłużają się do czterech-pięciu. Przy nauce i zabawie, na jakie pozwala Logo, uczniowie zapominają o kolacji...

Piszą krótkie, efektowne programy, wykazując zadziwiającą obojętność do gier. Stosowaliśmy już mikrokomputer na lekcjach fizyki i matematyki z wykorzystaniem oprogramowania profesjonalnego do rozwiązywania układów równań czy rysowania wykresów funkcji. Zorganizowaliśmy turniej szachowy

▶ 6

5 ◆

ze Spectrum. Poszukuję sponsora, który sfinansowałby mi zakup jeszcze kilku egzemplarzy i drukarki...".

Ewa Bodzioch, LO Chrzanów, woj. katowickie:

"Mamy dwa komputery, ale chcielibyśmy urządzić większą pracownię. Są środki, ale największy kłopot jest z zakupem monitorów. Czy byłoby możliwe, aby pewne ich ilości zostały przeznaczone tylko dla potrzeb szkół?"

W jednym z pokojów piotrkowskiego akademika pogrążony w pracy przy edytorze tekstów Stanisław Młynarski (Zespół Szkół Zawodowych Nr 1 w Myszkowie, woj. częstochowskie) chce wydrukować instrukcje języka Logo dla uczniów:

"Prowadzę od pół roku koło informatyczne na bazie jednego Spectrum. Wykorzystując Basic i Logo, które poznałem w czasie nauki w studium, staramy się rozwiązywać z uczniami różne problemy matematyczne i fizyczne, nie rezygnując z konstruowania własnych programów dydaktycznych i sprawnościowo-logicznych. Myślę, że od września rozpoczniemy zajęcia przy pięciu stanowiskach. Na co narzekam? Na zupełny brak ciekawej, fachowej literatury dostosowanej do naszych potrzeb".

Bożena Szpytna, LO w Prudniku, woj. opolskie:

"W czasie ferii zimowych otrzymaliśmy kilka Spectrum i Meritum. Od kwietnia będę prowadzić koło informatyczne i z tego co wiem, zainteresowanie będzie olbrzymie.

Myślę, że pomogą mi "Bajtek" i "Komputer". Z Wrocławia otrzymaliśmy dobre programy dydaktyczne z fizyki. Na razie będą punktem wyjścia do tworzenia własnych. Poza tym nawiązałam w Piotrkowie wiele znajomości i liczę, że będziemy wymieniać programy".

Józef Wyspiański, LO w Lubinie, woj. legnickie:

"Z powodzeniem wykorzystujemy nasze trzy Spectrum, przede wszystkim na zajęciach z fizyki. Programy tworzone zarówno przez uczniów, jak i nauczycieli służą uzupełnieniu procesu lekcyjnego. Bardzo dobrze spisują się komputery w prezentacji danych pomiarowych, szacowaniu błędów i analizie obliczeniowej doświadczeń. W zasadzie nie widzimy przeciwwskazań, aby wypożyczać sprzęt do pracy w domach".

Zastosowania mikrokomputerów w szkołach oscylują, jak widać, wokół matematyki i fizyki, co nie jest wcale bez znaczenia, jeśli zwrócimy uwagę na niepopularność tych kierunków studiów wśród absolwentów szkół średnich. Być może dzięki inwencji pedagogów wzrośnie zainteresowanie deficytowymi kierunkami. Nadarza się okazja...

Henryk Białek, metodyk przy ODN w Krakowie:

"Dobrze by było, gdyby elementy informatyki prowadził pedagog ze szkoły. Chodzi o ścisły kontakt tego przedmiotu z innymi. Jest to bowiem duża szansa dla przedmiotów ścisłych, które w wyniku reformy utraciły sporo godzin i muszą szukać nowych sposobów przyciągnięcia młodzieży. Z tego względu informatyka w szkole oprócz swojej roli odrębnej, nazwijmy ją zawodowej, pełniłaby funkcje uzupełniające w odniesieniu do innych przedmiotów nauczania. Mikrokomputer bowiem okazał się potężnym środkiem dydaktycznym i z powodu swej uniwersalności powinien znaleźć się w wykazie obowiązkowych pomocy naukowych. W Krakowskiem jest już wiele placówek, które posiadają ten sprzęt. O wielkim zainteresowaniu niech świadczy fakt, że na kolejną edycję Studium Informatyki złożyło podania ponad 50-ciu pedagogów z województwa. Rzecz jasna nie dla wszystkich starczy miejsca. W swojej pracy nie jesteśmy sami. Z pomocą szkołom przychodzą uczelnie: Politechnika Krakowska i Akademia Górniczo-Hutnicza. Bez przerwy są propozycje współpracy, szczególnie jeśli chodzi o literaturę informatyczną, co jest dla nas bardzo istotne.

Zaczęło się od Szkoły Podstawowej Nr 131, gdzie pod okiem p. Ciesielskiego wykorzystuje się komputery do indywidualnego nauczania dzieci specjalnej troski.

W różnych placówkach jest dużo pokazów i zastosowań. Moje macierzyste X LO ma cztery Spectrum, lecz prawdziwej pracowni nie posiadamy ze względu na brak sali. Komputery są wykorzystywane raczej na zajęciach pozalekcyjnych. Przekonaliśmy się, że bardzo opłaca się wyszkolić uczniów na programistów, gdyż tworzą rzeczywiście interesujące rzeczy. Obecnie staramy się prowadzić zajęcia również dla nauczycieli, gdzie sprawdzamy różne koncepcje dydaktyczne, przygotowując się do wprowadzenia informatyki do programu zajęć szkoły. Marzymy o ułatwionym zakupie czystych kaset magnetofonowych o czasie odtwarzania do 15 minut".

Zastanawia rzeczywiście fakt, jak uczniowie godzą naukę w szkolnej ławie z czasochłonnym procesem budowania programów po lekcjach. Z doświadczenia wiem, że odbija się to na wynikach klasyfikacji. Z drugiej strony nie wolno im tego zabronić, ale oby cena, jaką zapłacą, nie była zbyt duża. Czy jednak programy dydaktyczne powinny powstawać w szkołach? Czy tylko nauczyciel potrafi stworzyć dobry program dydaktyczny? Myślę, że kadra nauczycielska w większości nie będzie dysponować na to czasem – i tak już dziś (wbrew pozorom) mocno go brakuje na cele wychowawcze.

KRZYSZTOF MAMCARZ

JAK TWORZYĆ PROGRAMY EDUKACYJNE?

Z niecierpliwością czekamy na dzień, w którym dla większości naszych Czytelników od instrumentu ważniejsze stanie się grane na nim dzieło, od modelu komputera – zawarta w jego oprogramowaniu myśl dydaktyczna, artystyczna lub naukowa.

Jak dotychczas o tym, gdzie i przez kogo, na jakich zasadach będą tworzone i rozprowadzane polskie programy masowego użytku, wiadomo nawet mniej, niż o tym, jakie będą przyszłe polskie komputery. Pisaliśmy już, że nasza redakcja ma w tym zakresie własne plany i ambicje. Nie jesteśmy, na szczęście, jedynymi w naszej trosce o polskie programy.

Oto wstęp i dwa krótkie fragmenty dalszej części dokumentu przygotowanego w Pracowni Szkolnych Zastosowań Informatyki Instytutu Badań Pedagogicznych na posiedzenie Zespołu Doradczego Ministra Oświaty i Wychowania ds. Informatyki. Nie znamy jeszcze stanowiska resortu oraz zespołu, poglądy autora – miejscami odbiegające od utartych opinii – wydają się nam jednak godne szerszego przedstawienia.

Redakcja

Nie ma jeszcze wzorów ani jasnej koncepcji użytkowania komputerów w procesie nauczania, lecz każdy nauczyciel, po opadnięciu emocji związanych ze zdobywaniem wymarzonego sprzętu niecierpliwie oczekuje na pojawienie się programów, w nadziei, że będą one rewelacyjną i uniwersalną pomocą dydaktyczną. To zapotrzebowanie wywołuje żywiołowe działania mające na celu wypełnienie luki. W chwili obecnej krąży po kraju pewna, bliżej nieokreślona liczba programów, które można zaliczyć do następujących grup:

PROGRAMY OFEROWANE PRZEZ FIRMY PRYWATNE

Programy te są opracowywane przez te firmy we własnym zakresie lub kopiowane ze źródeł zagranicznych. W obu przypadkach jednak są to programy nie oparte o żadną koncepcję dydaktyczną, tworzone w oderwaniu od szkół i ośrodków dydaktycznych. Przykładem może być tu program "AKUSTYKA I MAGNETYZM" oferowany za cenę około 10 tys. złotych. Jest to prawdopodobnie tłumaczone z niemieckiego, niezręczne przeniesienie do pamięci komputera złego tekstu podręcznikowego, pełne błędów merytorycznych i szkodliwe metodycznie. Inna firma oferuje w podobnej cenie pakiety programów, będących tłumaczeniami programów brytyjskiej firmy Longman. Są one poprawne merytorycznie i programowo, natomiast budzi wątpliwości ich wartość dydaktyczna. Należy pamiętać, że mikrokomputer ZX Spectrum nie znalazł szerszego zastosowania w szkołach brytyjskich. Pełni on funkcję pomocniczą, używany jest do ćwiczeń czy samodzielnych zajęć uczniowskich, a w domu ucznia – jako "korepetytor" pomagający w powtarzaniu materiału. W poważnych ośrodkach dydaktycznych jest on niemal nie znany – brak też poważnie opracowanych programów. Przysłowiowym "kotem w worku" jest pakiet programów, które "wychodząc naprzeciw społecznym zapotrzebowaniom" oferuje za 99.500 firma POLMER. Nierozważne byłoby kupowanie takiego pakietu przed dokonaniem weryfikacji i oceny jego przydatności. Zawarte w ofercie uwagi na temat Nauczania Wspomaganego Komputerowo powtarzają, zapewne w dobrej wierze, opinie o rzekomym urzeczywistnieniu za pomocą komputerów „pełnej i kontrolowanej" indywidualizacji nauczania.

Nauczanie a komputer – ustalenie obszaru wpływu nowego środka na cele i realizację celów nauczania.

- fakty i dane .... wywoływanie,
- związki i relacje .... zrozumienie,
- umiejętności praktyczne .... stosowanie,
- rozwiązywanie problemów .... analiza,
- kreowanie twórczych postaw .... synteza.

Uczenie się a komputer – ustalenie obszaru wpływu nowego środka na metodykę nauczania.

- na czym polega działanie uczącego się?
- jak to, co robi uczący się, pomaga mu w nauce?
- czy wszyscy w grupie uczą się w taki sam sposób i w podobnym tempie?
- czy nowy środek może w tym pomóc?

PROGRAMY PRZYWOŻONE Z ZAGRANICY INDYWIDUALNIE

Programy te kupowane są przez niefachowców, w przypadkowych źródłach, tak że ich wartość i przydatność dla nauczania jest wątpliwa.

Różnorodność czynników wpływających na jakość programu edukacyjnego powoduje, iż dobre programy są wynikiem pracy wieloosobowych zespołów specjalistów różnych dziedzin. Jednak w zespołach takich rolę wiodącą powierza się dydaktykom przedmiotowym.

PROGRAMY PISANE PRZEZ NAUCZYCIELI I UCZNIÓW

Napisanie każdego programu komputerowego wiąże się zawsze z rozwiązaniem pewnego problemu – ma zatem zawsze wartość kształcącą. Natomiast ten sam program stosowany w nauczaniu może zamiast korzyści przynieść szkody, jeśli w miejsce problemu pokaże się uczniom jego gotowe rozwiązanie. Wywołuje to niebezpieczne zmniejszanie się aktywności uczniów w wyniku wprowadzenia komputerów do klasy. Skuteczne napisanie programu staje się źródłem uzasadnionej dumy i radości dla jego twórców. Z drugiej strony brak możliwości poznania dobrych wzorów, brak literatury i dyskusji powoduje, że wśród twórców programów nie wyrabia się samokrytycyzm i umiejętność oceny własnej pracy. Powstają prace o charakterze dyletanckim. Nauczyciele pełni dumy z prac swoich czy swych uczniów chcieliby, żeby zostały one rozpowszechnione, nie zdając sobie sprawy z ich obiektywnej wartości. Tak więc

PRAKTYCZNIE NIE MA OBECNIE W KRAJU PROGRAMÓW EDUKACYJNYCH.

Szkoła jednak nie może czekać! Gdy zdobędzie komputer, uczniowie, rodzice, dyrekcja, słowem wszyscy domagają się od nauczyciela, aby go wykorzystywał. Nauczyciel więc musi za wszelką cenę znaleźć jakikolwiek program. Wszyscy zatem rozumieją, że "coś trzeba zrobić". Pod adresem resortu oświaty formułowane są żądania, żeby przyspieszyć pojawienie się większej liczby programów np. przez organizowanie konkursów. Zdaniem niektórych ministerstwo powinno zakupywać programy (wartościowe?) i kierować je do produkcji (gdzie?). Powinno się zorganizować wytwórnię programów, powinno się dotować programy i ich twórców. Programy powinny być dostarczane do szkół bezpłatnie. Itd., itp.

Doświadczenie (i to bardzo kosztowne) innych krajów uczy, że opracowanie wartościowego oprogramowania edukacyjnego

Typowy proces powstawania programu dydaktycznego przebiega przez następujące etapy:

A. Powstanie pomysłu

Pomysł programu pojawia się w związku z określoną sytuacją dydaktyczną. Np. ważne zagadnienie składania drgań harmonicznych prowadzące do syntezy drgań o dowolnym kształcie jest trudne do wprowadzenia w szkole średniej ze względu na konieczność prowadzenia żmudnych obliczeń; wykorzystanie komputera pozwala z powodzeniem trudności te ominąć.

Pomysł programu powinien być przedstawiony grupie złożonej z dydaktyków przedmiotowych, nauczycieli i programistów.

(Nawiasem mówiąc, czasem zdarza się, iż nowe możliwości, które stwarza komputer, powodują zmianę doboru treści kształcenia a stąd także nowe sytuacje dydaktyczne).

B. Opracowanie wstępnego scenariusza programu

C. Ocenę scenariusza programu z punktu widzenia przedmiotowo-dydaktycznego (wymiar merytoryczno-przedmiotowy) oraz z punktu widzenia informatycznego (wymiar komputerowy).

D. Napisanie programu

E. Sprawdzenie programu przez nauczycieli pracujących w grupie

F. Modyfikacje scenariusza i programu

G. Testowanie programu w klasie przez innych nauczycieli

H. Ostateczne korekty programu

I. Napisanie instrukcji metodycznej oraz dokumentacji programu

J. Przekazanie programu do produkcji

możliwe jest tylko przy równoczesnym rozwijaniu całościowej koncepcji dydaktycznej i krytycznym spojrzeniu na program nauczania danego przedmiotu. Nie mając kryteriów oceny, ani możliwości selekcji, nie powinno się ulegać naciskom w celu

➧ 44



CZASOPISMO DLA WŁAŚCICIELI KOMPUTERÓW

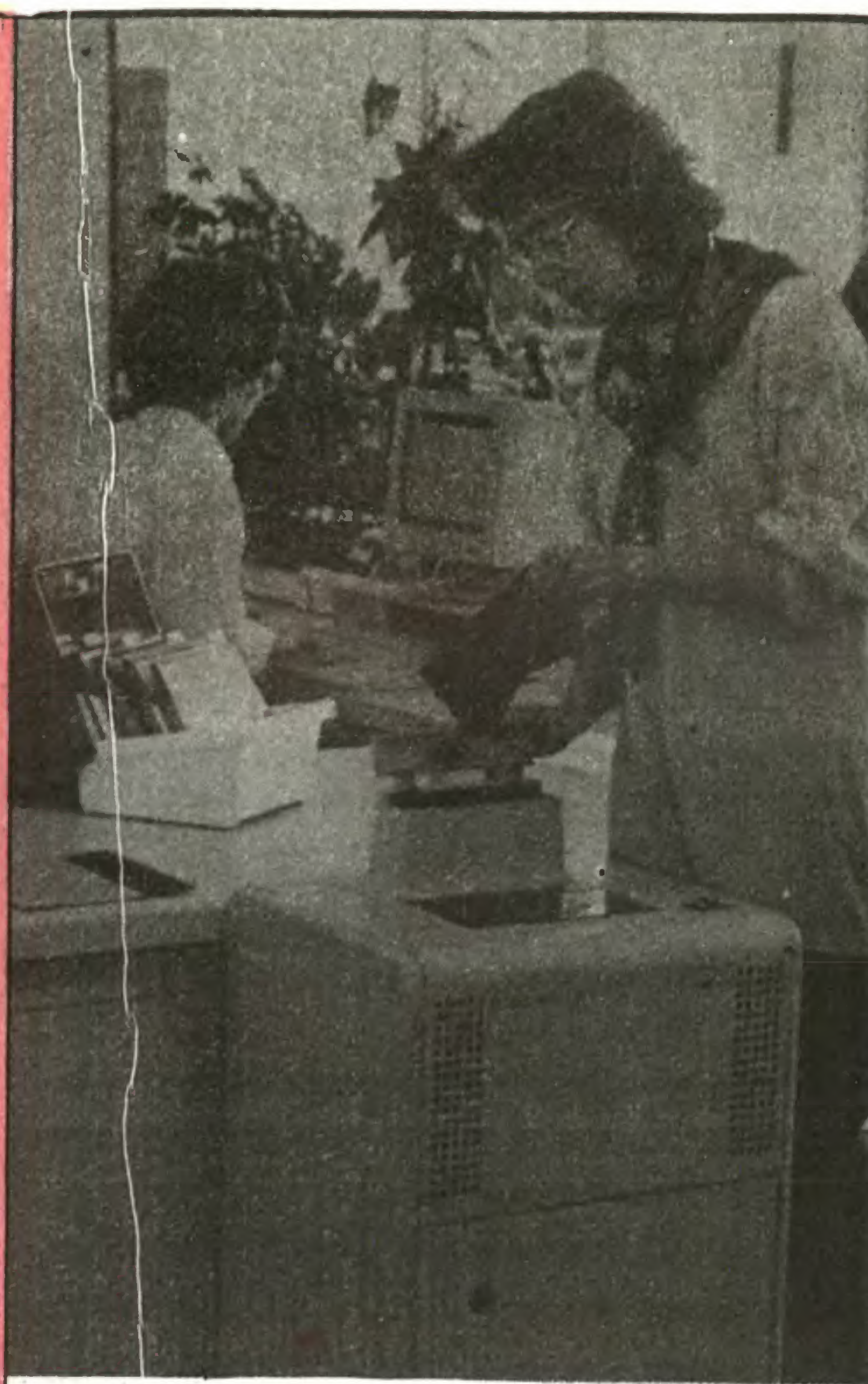
Mimo powszechnego wykorzystywania w informatyce magnetycznego, a ostatnio także optycznego sposobu zapisu, pozycja druku jako najpopularniejszego na świecie sposobu utrwalania i przekazywania informacji pozostaje niezagrożona. Niektórzy wydawcy czasopism przeznaczonych dla informatyków coraz częściej jednak próbują wprowadzić bardziej atrakcyjną dla posiadaczy komputerów formę przekazu – zapis na kasecie lub dyskietce. Początkowo miało to być jedynie uzupełnienie tradycyjnej gazety – dyskietka była jedynie "załącznikiem" do każdego numeru czasopisma. Ostatnio jednak pojawiły się także czasopisma wydawane wyłącznie na dyskietkach. Przykładem takiego wydawnictwa jest amerykańskie czasopismo Up Time, wydawane przez Viking Technologies Inc. Jest ono przeznaczone dla posiadaczy Apple II, ale wydawca zapowiedział już rozszerzenie oferty także na posiadaczy IMB PC i Macintosha. Up Time można kupować zarówno w punktach zajmujących się sprzedażą „zwykłych" czasopism jak i zaprenumerować. Czasopismo na dyskietkach przynosi wydawcy zupełnie niezły dochód – w tym roku jest spodziewane osiągnięcie obrotów przekraczających milion dolarów.

(gs)

CIĘŻKIE CZASY DLA SZPIEGÓW GOSPODARCZYCH?

Dostanie się niektórych informacji w niepowołane ręce może narazić firmę na bardzo poważne straty. Coraz więcej ośrodków obliczeniowych, laboratoriów badawczych a także zwykłych biur posługuje się więc różnego rodzaju urządzeniami służącymi do dokładnego niszczenia notatek, dokumentów i nośników magnetycznych. Przykładem takiego urządzenia może być przyrząd H5 oferowany przez firmę Siemens. Umożliwia on szybkie pocięcie na niewielkie kawałki i dokładne wymieszanie nie tylko kartek papieru, ale także kaset z taśmą pochodzących z maszyn do pisania i drukarek oraz dyskietek i kaset komputerowych. Urządzenie spełnia wymagania dotyczące dopuszczalnej wielkości pociętych kawałków określonej w normie DIN 32757. Siemens oferuje całą grupę tego typu urządzeń do niszczenia dokumentów. Najmniejsze z nich ma wymiary 220x234x155 mm i w ciągu 10 sekund może pociąć 2 kartki papieru formatu A4 (każdą z nich na 6200 kawałków o wymiarach 0,8x12,5 mm). Dla centrów obliczeniowych przeznaczone jest urządzenie oznaczone symbolem H4. Na pocięcie na kawałki o wymiarach 2,2x24 mm wstęgi papieru (wydruku) o długości 12 m i szerokości 374 mm potrzebuje ono 1 min.

(gs)



JUŻ SERYJNIE

Na początku tego roku firma Toshiba rozpoczęła w swym zakładzie w Oita seryjną produkcję układów pamięci półprzewodnikowych 1 MB. W kwietniu miesięczna produkcja miała osiągnąć milion sztuk. Także Matsushita od lata tego roku produkować będzie seryjnie pamięci 1 MB. Miesięczna produkcja zakładów Uozu wynosić będzie 100 000 szt. Rozpoczęcie wielkoseryjnej produkcji spowoduje prawdopodobnie w najbliższym czasie dalszą obniżkę cen pamięci 256 KB (gs)

GDY ZAWIEDZIE SIEĆ

Chwilowy zanik napięcia w sieci zasilającej prowadzi najczęściej do poważnych zakłóceń w pracy systemu komputerowego, a tworzone właśnie programy lub wyniki prowadzonych obliczeń tracone są bezpowrotnie. Amerykański koncern Elgar oferuje użytkownikom komputerów osobistych IBM (PC, XT, AT) system FailSafe, pozwalający wykluczyć powstanie szkód na skutek przerw w zasilaniu. Urządzenie składa się z baterii bezobsługowych akumulatorów, falownika, stacji dysków i mikroprocesorowego układu sterującego współpracującego z DOS. W wypadku przerwy w zasilaniu następuje przejście na zasilanie z akumulatorów (poprzez falownik). Gdy jednak przerwa jest dłuższa niż 4 min., FailSafe zapisuje na dysku zawarte w pamięci RAM informacje i wyłącza się. Z chwilą przywrócenia napięcia w sieci następuje automatyczne "władowanie" utrwalonych na dysku informacji do komputera. (gs)

CO TO BYŁ ZA WEEKEND...!

Długo maklerzy na londyńskiej giełdzie będą pamiętać piątek 14 marca. Tego dnia bowiem odmówił posłuszeństwa system elektronicznej rejestracji transakcji. Ręczne wystawianie odpowiednich dokumentów księgowych i prawnych, umów kupna-sprzedaży itp. trwało jeszcze w niedzielę. Aby nie mieć kłopotów z resztówkami (większość jednak wyczyszczono), wyznaczono dwie firmy, które po paru tygodniach wykaraskają się z pozostałości.

Giełda zafundowała sobie ten system zaledwie dwa lata wcześniej, kosztem 42 mln funtów szterlingów. Na giełdzie chodzi o pieniądze, należy więc przypuszczać, że przezorni maklerzy i funkcjonariusze giełdy dokładnie zbadali sprawę zarówno sprzętu, jak i samego oprogramowania nie tylko pod kątem prawidłowości identyfikacji stron transakcji i księgowań, ale także niezawodności. Giełdy całego świata są dość sceptycznie nastawione do nowinek, bowiem robienie pieniędzy na ułamkach procentów od obracanych sum jest sprawą zbyt poważną, aby można było tu eksperymentować.

W piątek zdarzyło się, że ze względu na bardzo ożywione obroty zdecydowano się rozładować tłok przez dodanie do sieci dwu mikrokomputerów. Były one już używane i sprawdzone na giełdzie. Dlatego przed włączeniem nie przeprowadzono osobnej inspekcji technicznej. Ale gdy je włączono – nastąpiło sprzężenie zwrotne. Zrobiła się pętla, program stale gadał sam do siebie i nie można było tego przerwać – tzn. wprowadzić lub wyprowadzić informacji – bez wyłączenia całego systemu. Najpierw podejrzewano najgorsze, mianowicie że coś się złego stało z podsystemem gromadzenia i transmisji danych. Oznaczałoby to niezwykle pracochłonne wyszukiwanie błędu. Ale i tak cztery godziny pracowała własna ekipa techniczna, zanim błąd zlokalizowano i stwierdzono, że to właśnie dołączenie dwu dodatkowych mikrokomputerów spowodowało zapętlenie się systemu.

Giełda wyciągnęła z tego pożałowania godnego przypadku nauki. Teraz, aby zawsze można było robić pieniądze, żaden mikrokomputer nie może być podłączony do systemu bez uprzedniej kontroli technicznej. Nawet między już sprawdzonymi komputerami i resztą systemu trzeba umieścić "filtry", które uniemożliwią wadliwie działającej maszynie zniszczenie reszty systemu.

Operacje giełdowe są trudne do skomputeryzowania, bowiem w grę wchodzi wielka liczba stron transakcji. Może też być bardzo wiele przedmiotów tych transakcji tj. rodzajów papierów wartościowych. Ceny zmieniają się co minuta nieledwie. Płatności dokonują, w imieniu firm maklerskich, liczne banki, a sami maklerzy działają w imieniu mnóstwa klientów, którzy też mają swoje banki.

Giełda londyńska wyda jednak na uzupełnienie systemu komputerowego 38 mln funtów w tym roku, co wcale nie jest jeszcze dużo. Wielkie amerykańskie firmy finansowe (niekoniecznie banki) potrafią wydawać do 100 mln dolarów rocznie na systemy komputerowe, co jednak wcale nie chroni ich przed paskudnymi wpadkami. Jeśli bowiem uwzględnić, że w wielkim banku dealerzy walutowi obracają dziennie ok. 10 mld dolarów, zarabiając na tym około 10 mln, to skutki finansowe wstrzymania operacji przez awarię, choćby na dwie godziny, muszą być bardzo dotkliwe dla akcjonariuszy banku. (JAL)

gazeta NA MIARĘ

Na międzynarodowym seminarium PRESSINFO (Forum Prasy i Informacji) w Walencji w 1985 r. toczyła się dyskusja na temat „Informatyka a środki masowego przekazu". Z tej dyskusji, opublikowanej w nr. 10/1985 kwartalnika "Agora", wybraliśmy dziś wypowiedź Johanna Fritza, redaktora naczelnego wiedeńskiej "Die Presse", wiceprzewodniczącego Zrzeszenia Austriackich Wydawców Gazet, mówiącą o tym, jak wielkie zmiany w sposobie redagowania i wydawania prasy spowodowała elektronika.

W najbliższym czasie zamieścimy inny fragment dyskusji, w którym będzie mowa m.in. o społecznych konsekwencjach rozwoju nowych środków komunikowania, a także – wynikających z tego zagrożeniach...

Zainstalowanie pierwszego skomputeryzowanego systemu składu przez Oklahoma Publishing Co. w 1962 roku rozpoczęło szybkie wkraczanie komputerów do drukarń. W Europie elektroniczne przetwarzanie danych przedostało się do przemysłu poprzez działy księgowości, płac czy też działy kolportażu. Przetwarzanie tekstu, czyli elektroniczne składanie i redagowanie, przyjęło się znacznie później. Wielostronny rozwój systemów informatycznych doprowadził do elektronicznego redagowania tekstów, a to ze swej strony umożliwiło pełne zintegrowanie działów: redakcyjnego, produkcji, kolportażu, ogłoszeń i administracyjnego w jeden organizm wydawniczy. Cóż może taki kompleks redakcyjno-wydawniczy uczynić?

- Elektroniczny system redagowania tekstu i składania jest znacznie szybszy od konwencjonalnych metod przygotowania tekstów do druku. Redakcyjny etap prac może zostać w gazecie przedłużony i może być włączany do procesu produkcyjnego znacznie później, bez stwarzania kłopotów dla składu i łamania. Tym samym kolumny gazet mogą pozostawać "otwarte" rzeczywiście do ostatniej chwili czekając na najświeższe wiadomości, co pozwala stać się gazetom bardziej konkurencyjnymi w stosunku do tak zwanych "szybkich", elektronicznych środków masowego przekazu (radio, TV, gazety komputerowe typu "teletext", które można oglądać na ekranie domowego telewizora – przyp. tłum.).
- Wprowadzenie terminali komputerowych u korespondentów otworzyło im całkowicie nowe możliwości, pozwalając na transmitowanie tekstu w ostatniej chwili liniami telefonicznymi.
- Wiadomości agencyjne nadawane bardzo szybkimi metodami mogą teraz być swobodnie otrzymywane drogą połączenia telefonicznego po nakręceniu kodu agencji lub też kodu agencji i numeru wybranej informacji. Serwis depesz agencyjnych może przy tym być odbierany albo w formie taśmy teleksowej, albo też przekazywany bezpośrednio na monitor terminalu, do banku informacji komputera redakcyjnego, czy też wprost do urządzenia elektronicznego składu.
- Elektroniczna ciemnia, elektroniczne kamery itp. ułatwiają szybkie publikowanie zdjęć w gazetach. Zdjęcia, jak również kolor, mają wysoką wartość informacyjną i przyciągają czytelników.
- Jakość prac redakcyjnych zależy w dużej mierze od dostępności danych o charakterze archiwalnym lub encyklopedycznym. Obecnie łatwo jest magazynować na taśmie magnetycznej każdą notkę czy artykuł, jakie ukazały się w gazecie.
- Rozwój telekomunikacji i technologii banku danych daje wiele możliwości współpracy międzyredakcyjnej, czego efektem jest poprawa jakości gazety. Dla przykładu: kilka gazet lokalnych może razem stworzyć wspólny bank informacji o charakterze ogólnokrajowym czy światowym!
- Nowe technologie pozwalają na niesłychaną wprost możliwość mutowania gazety i zmieniania jej zawartości w zależności od geograficznych bądź demograficznych wymagań. Prowadzi to do koncepcji "gazety na miarę", czyli gazety "skrojonej" wedle indywidualnych gustów czytelnika.
- Przekazywanie fascimille całych stron do odległych drukarń tworzy nowe możliwości usprawnienia kolportażu.

Zmiany w tradycyjnych formach wydawania gazet doprowadziły do poważnych zmian w codziennej pracy redakcyjnej oraz do dramatycznych redukcji personelu. Istnieją obawy, że elektroniczne technologie zamienią dziennikarzy w techników lub oderwą ich od najważniejszych obowiązków profesjonalnych. Przed przestawieniem się na komputeryzację wydawcy muszą bardzo wnikliwie przeanalizować charakterystykę prac redakcyjnych tak, by zmiana stylu pracy nie doprowadziła do obaw czy niezadowolenia i dała maksimum korzyści.

Nowe technologie, racjonalizacja i ulepszenia zwiększające wydajność pracy zawsze są źródłem kłopotów społecznych. Powstaje problem zapewnienia gwarancji zachowania miejsca pracy czy też zastępowania ludzi przez maszyny. W kilku krajach europejskich wprowadzenie automatyki i nowych metod pracy zostało opóźnione, a czasem nawet zablokowane do pewnego stopnia przez związki zawodowe. Liczba zagrożonych miejsc pracy w gazecie jest znaczna: zecerzy-linotypiści i składacze ręczni, metrampaże, maszynistki, korektorzy, retuszerzy, pracownicy chemigrafii, archiwiści itp. Głębokie nieporozumienia i nieufności między kierownictwem a pracownikami i związkami zawodowymi mogą być zminimalizowane, jeśli program automatyzacji będzie przedstawiony nie tylko pod kątem aspektów technicznych, ale również społecznych. Należy też przedstawić odpowiednie propozycje rozwiązania kłopotów, np. plany wcześniejszego przechodzenia na emeryturę lub przeszkolenia pracowników w nowych lub istniejących zawodach potrzebnych gazecie. Nowe technologie stwarzają bowiem również nowe miejsca pracy. Gazety będą potrzebować ludzi obsługujących i konserwujących skomplikowane systemy komputerowe. Będą potrzebować ludzi zbierających i redagujących wiadomości, ludzi zbierających reklamy i sprzedających gazetę.

Zlanie się w jedno technologii przetwarzania, przechowywania i przekazywania informacji prowadzi do rozwoju nowych środków komunikowania, które otwierają przed wydawnictwami znakomite perspektywy poszerzenia ich rynku i pełniejszego wykorzystania posiadanych już umiejętności zbierania i kolportowania informacji. Można sobie wyobrazić gazetę dostarczaną drogą kablową, radiową lub satelitarną wprost na ekran domowego telewizora czytelnika, który – mając podłączone urządzenie drukujące – będzie mógł, jeśli będzie chciał, sam wydrukować sobie całość lub interesujący go fragment gazety. Drogą skomputeryzowanego systemu przekazywania informacji czytelnicy będą też w stanie zamówić, poprzez domowe terminale, dowolny artykuł, jaki pragną przeczytać.

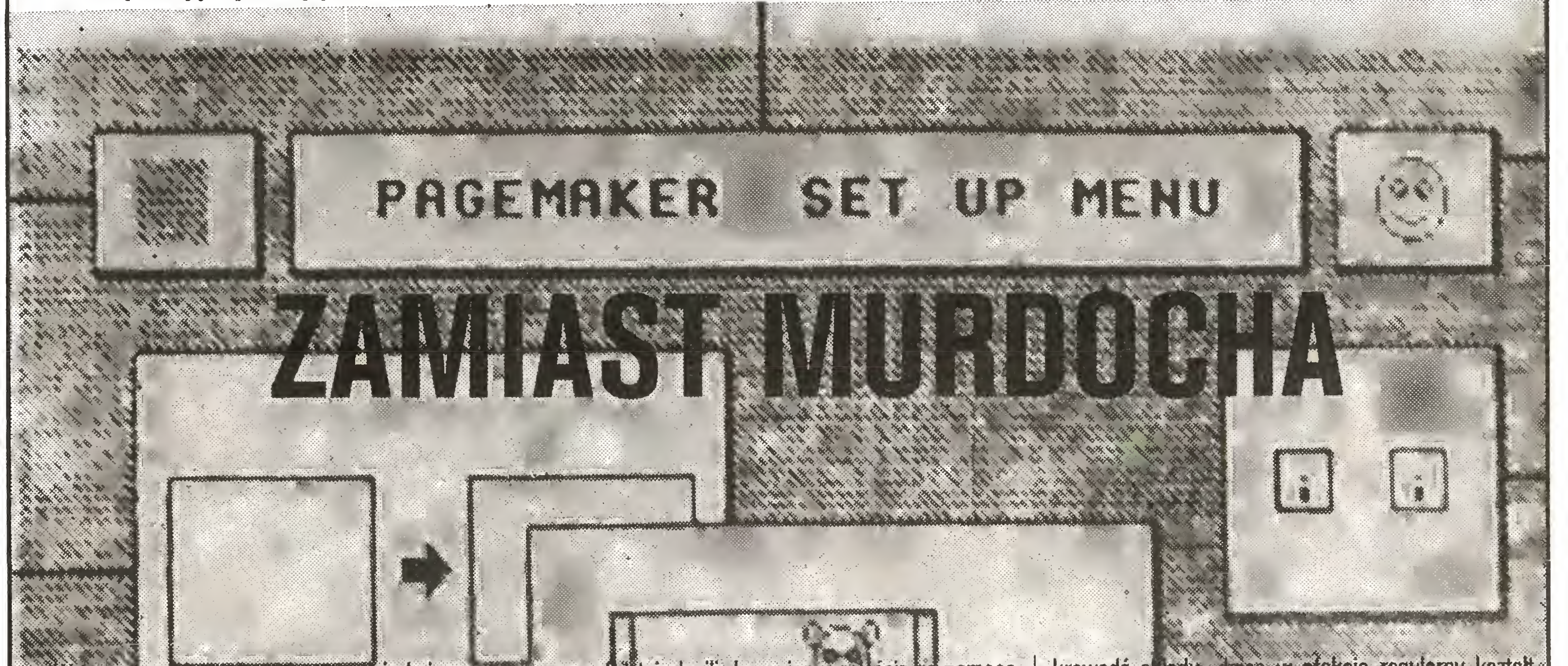
Nie będzie natomiast poważniejszych zmian zarówno w organizacji, jak i w sposobie prowadzenia gazety. Kwestie, w jaki sposób informacja została zdobyta, czy też w jaki sposób została wykorzystana, będą w dalszym ciągu regulowane przez działania prawne lub ustawowe oraz poprzez siły społeczne.

Tłumaczył i opracował R. B.

Od tłumacza:

Ten tekst, zamiast być pisany "na brudno", przepisywany na maszynie i dostarczony "człowiekiem" do redakcji, powinien zostać wystukany na klawiaturze komputera, następnie przeredagowany i poprawiony na tymże i przekazany linią telefoniczną do pamięci komputera redakcyjnego.

Dokonał się akt socjalnego bestialstwa. Wydawniczy potentat, Rupert Murdoch, wyrzucił na bruk sześć tysięcy ludzi, redaktorów, techników i drukarzy. Kolebka londyńskiej prasy, okolice Fleet Street, padła ofiarą nowej zarazy, nowych mikrobów – mikroprocesorów. W dzielnicy doków Murdoch otworzył całkowicie skomputeryzowaną fortecę: nowoczesny kompleks redakcyjno – poligraficzny. Można by zacytować Leca: „Gdy warsztat przerośnie formę, dzieją się rzeczy potworne".



ZAMIAST MURDOCHA

W fortecy w Wapping wdzięcznie ćwierkają word-procesory, dyskretnie szumią maszyny drukarskie, a jedyny papier, z jakim ma się do czynienia, to notes dziennikarza i finalny produkt – gazeta. Pisanie, redagowanie, korekta, obieg informacji i dostęp do źródeł – wszystko na ekranach. Ani jednej maszyny do pisania, tylko edytory tekstu, modemy, światowa sieć i satelity. Rodzi się nowy gatunek dziennikarza z mikrokomputerem za pazuchą. Oczywiście zdjęcie także można przesłać do firmy przez telefon. Nie warto już wspominać o etapie makietowania pisma i przygotowania form drukarskich. To byłaby bajka z tysiąca i drugiej nocy. Jeśli zaś tak ma wyglądać przyszłość, to nie ominie nikogo, spokojna głowa. Tak bowiem jest urządzony ten światek, że jak w przysłowiowym Pernambuco ktoś na przykład wykombinuje światło wyciskane z żuby, to z całą pewnością zapragną je mieć w Parzęczewie.

Co zamiast Murdocha? Jest, na szczęście, wiele propozycji przekształcenia trywialnego biurka w fantastyczną wytwórnię gotowych i kompletnych stron czasopisma, jedna po drugiej, z ilustracjami, zdjęciami, tytułami i całym tekstem. Wszystko w dowolnym kolorze, nasyceniu i formie. Prawdziwa kraina szczęścia dla grafików, a dla redaktorów niebywała okazja zapanowania nad całym, obecnie rozciągniętym do niemożliwości, etapem korekty i wielokrotnego przepisywania.

Dla popularnego komputera BBC stworzono znakomity program graficzny MAGAZINE MAKER. Jego poprzednik, oferowany przez firmę Mirrorsoft, nosił (nomen omen) nazwę FLEET STREET EDITOR. MAGAZINE MAKER stanowi system, na który składa się: AMX PAGEMAKER (dwa ROMy i dwa dyski 5,25 cala), Watford Electronics Video Digitiser (z własnym ROMem) i mysz AMX. Do pracy potrzeba jeszcze – prócz komputera – dwustronnie czytającego, 40/80-śladowego napędu dysków, kolorowej drukarki, magnetowidu, kamery video i dobrego monitora. Po przygotowaniu konspektu czynności i odpowiedniej liczby dysków można zaczynać.

Akcję wywołuje się przez naciśnięcie SHIFT-BREAK. W kilka sekund ukazuje się na ekranie główne menu. Od tej chwili dyrygujemy całością za pomocą myszy AMX. Popychamy kursor ku wybranemu ikonowi i naciskamy lewy skrajny przycisk myszy. Są trzy przyciski: WYKONUJ, RUSZAJ i KASUJ. Dwa pierwsze, dla uniknięcia ewentualnego zamętu, działają przemiennie. Menu pozwala zdefiniować liczne polecenia: wybierać materiał do wydruku, ustalać pozycję głównego ekranu itd., są opcje do korespondencji z dyskami i zmiany trybów dla transformacji plików grafiki (1-4 tworzenie, przechowywanie). Ustalamy parametry i wracamy do menu głównego. Sekcja szósta, to właśnie tworzenie grafiki rozumianej również jako kompozycja tekstu. U dołu ekranu znajdziemy serię ikonów, a na jego górnej krawędzi całą informację. Po wypisaniu słów WATFORD ELECTRONICS DIGITISER ekran jest gotów przyjąć całą grafikę. Z klawiatury można wnosić tylko niewielkie formy (tytuły itp.), dlatego program pozwala wiersze składu "makietować" metodą off-line, czyli rezerwując wstępnie powierzchnię pod tekst przez instrukcje WORDWISE lub VIEW. Za pomocą kursora ustala się okno, podobna opcja wyświetla się u góry i pojawia się krzyżyk, którym przez WYKONAJ i RUSZAJ precyzujemy położenie narożników okna. Kursor należy przenieść do ikonu tekstu i przyciskając mysz wywołuje się u szczytu ekranu nowe menu z szesnastoma (!) krojami pisma do wyboru, dostarczonymi na dysku krojów.

Trzeba teraz zdecydować o stopniu pisma i odległościach między wierszami. Możliwości mamy takie, jak najlepiej wyposażona drukarnia. Litery można zwężać i poszerzać wzwyż i wszerz oczka, swobodnie ustalać światła między wierszami i poszczególnymi literami. W zależności od liczby jednocześnie wykorzystywanych dysków, użytkownik ma możliwość przemiennie czerpać z dysku systemu, kroju lub tekstu. Mając krój i wielkość, wywołuje się kolejne wiersze po wskazaniu na selektorze u góry ekranu stosownej opcji. Kursor musi być ustawiony w aktualnym oknie, w którym tekst się rozpocznie. Naciskając WYKONAJ powoduje się wczytanie wierszy z dysku. Ekran pokazuje tekst wchodzący linia po linii, a MAGAZINE MAKER przerabia wiersze kolejno, tworząc lewą i prawą krawędź składu, dając w efekcie regularny kształt szpalty. Ekran może pokazać tylko część skończonej strony, więc kiedy tekst wypełni ekran lub aktualne okno, system będzie nakłaniał do wybrania rodzaju akcji poprzez opcje: kontynuowanie w aktualnej kolumnie (czyli całej stronie składu), przejście na następną lub określenie następnego okna. Siła systemu wzmocniona jest w dodatku przez jego wielkie możliwości wprowadzania video i łączenia go z pozostałymi elementami na jednej stronie. Watford Electronics Digitiser pozwala instalować je w kompozycjach MAGAZINE MAKER. Zgromadzone obrazy mogą być manipulowane przez software PAGEMAKER: łatwa operacja typu zoom na wybranej części obrazu, wzmacnianie go, przerzucanie z lewa na prawo lub obracanie górą do dołu, a także poddawanie "retuszowi".

MAGAZINE MAKER zawiera wszystkie możliwości niezbędne do realizacji stron tekstu i grafiki. Największą niedogodnością jest jednak obecna rozdzielczość drukarek punktowo-matrycowych. Jeśli w przyszłości spadną ceny drukarek laserowych, każdy użytkownik będzie mógł zrealizować projekt w doskonałej postaci.

Jesteśmy świadkami jakościowych zmian, jakich w edytorstwie jeszcze nie było. Oto za jednym zamachem można załatwić wszystkie problemy związane z procesem wydawniczym. Przedstawiony program jest jednym z wielu i dotyczy "tylko" pracy redakcji. Rewelacyjne możliwości, kultura pracy i najwyższy poziom efektu, a jeśli do tego odpowiednio wyposażyć drukarnię, to możliwość bezpośredniego odczytania projektu do wielobarwnego druku jest już taką szansą, że równa się prawdziwej rewolucji w stosunkach redakcja – drukarnia.

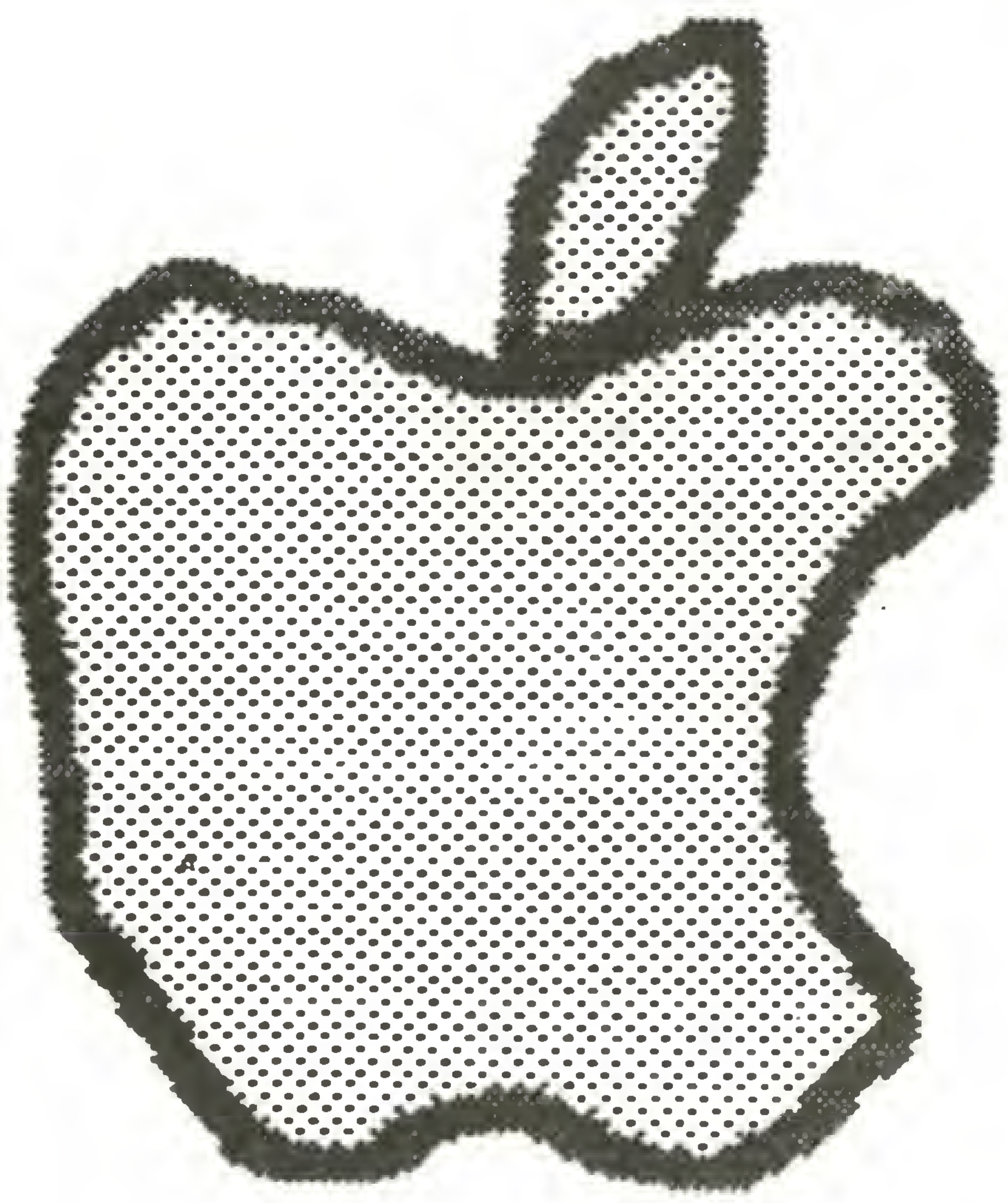
W przemyślanej polityce komputeryzowania poszczególnych dziedzin działalności wydawców kryje się piękna, nowa przygoda, nieporównana satysfakcja z udanego dzieła i okazja do rozwinięcia talentu. Słowem, nowy ROMantyzm w nowych RAMach.

STEFAN SZCZYPKA

Ps. Niech będzie wybaczony autorowi jego entuzjazm, albowiem rzecz całą podał do wiadomości właśnie grafik.

AaBbCcDdEeFfGgHhIiJjKkLlŁłMmNnOoPpQqRrSsTtUuVvWwXxYyZz1234567890.?!&$£[]

Mały Drukarz Każdy Wyd



Nie chcę nikogo namawiać do łamania przepisów, ale szansa osobistego napisania, złożenia, a na koniec wydrukowania tego, co się wymyśliło, jest pod ręką!!! Tylko od nas zależy, czy z tej okazji skorzystamy ...

Redakcja poprosiła mnie o zademonstrowanie możliwości przetwarzania tekstów przy pomocy komputera. Pomyślałem sobie, że jest to świetna okazja, by pokazać Czytelnikom także możliwość komputerowego składu.

Ci Czytelnicy, którzy kiedykolwiek pisali dla czasopism, wiedzą już, że tradycyjnie posyła się maszynopis (czasami jest to co gorsze niezbyt czytelny rękopis, który trzeba przepisać na maszynie). Redakcja opracowuje masznopis, robiąc tysiączne znaczki. Potem materiał ląduje w drukarni, gdzie jest (u nas często jeszcze ręcznie...) składany, czyli przetwarzany w druk. Potem próbne odbitki wracają do redakcji, gdzie panie, zwane korektorkami, poprawiają tzw. literówki, czyli błędy w składzie. Czasami autor dokonuje jeszcze niewielkich poprawek. Próbny druk jest następnie makietowany, czyli przy pomocy nożyczek redaktor techniczny stara się upchać materiał w objętości numeru. Trzeba też włamać wszystkie potrzebne rysunki, fotografie oraz akcenty graficzne. Materiały zmakietowane wracają teraz do drukarni, gdzie przygotowuje się matryce do druku. Potem jeszcze raz korekta, znowu podróż materiału do drukarni, by wreszcie po dwu miesiącach czytelnicy mogli przeczytać miesięcznik (w przypadku dzienników proces trwa jeden dzień, za to szata graficzna jest uboższa, zaś liczba błędów wielokrotnie większa). Poczem cały ten wysiłek wielu osób idzie na makulaturę, chyba że jest to bardzo ciekawy numer KOMPUTERA...

A można inaczej!

Krótki poradnik dla amatorów osobistego drukarstwa

czyli

aje Swoje Pismo

Bierzemy komputer, odpowiedni program i ... piszemy!!! Oczywiście piszemy na ekranie, wykorzystując możliwości graficzne komputera przemieszczamy teksty na stronach, dodajemy rysunki, zmieniamy krój czcionek, a nawet piszemy obcymi alfabetami:

fվրրւտճճւիուպաuղհգէթկ

йцукеягшщзфывапролд

$@å&¥®æµΩπʃø

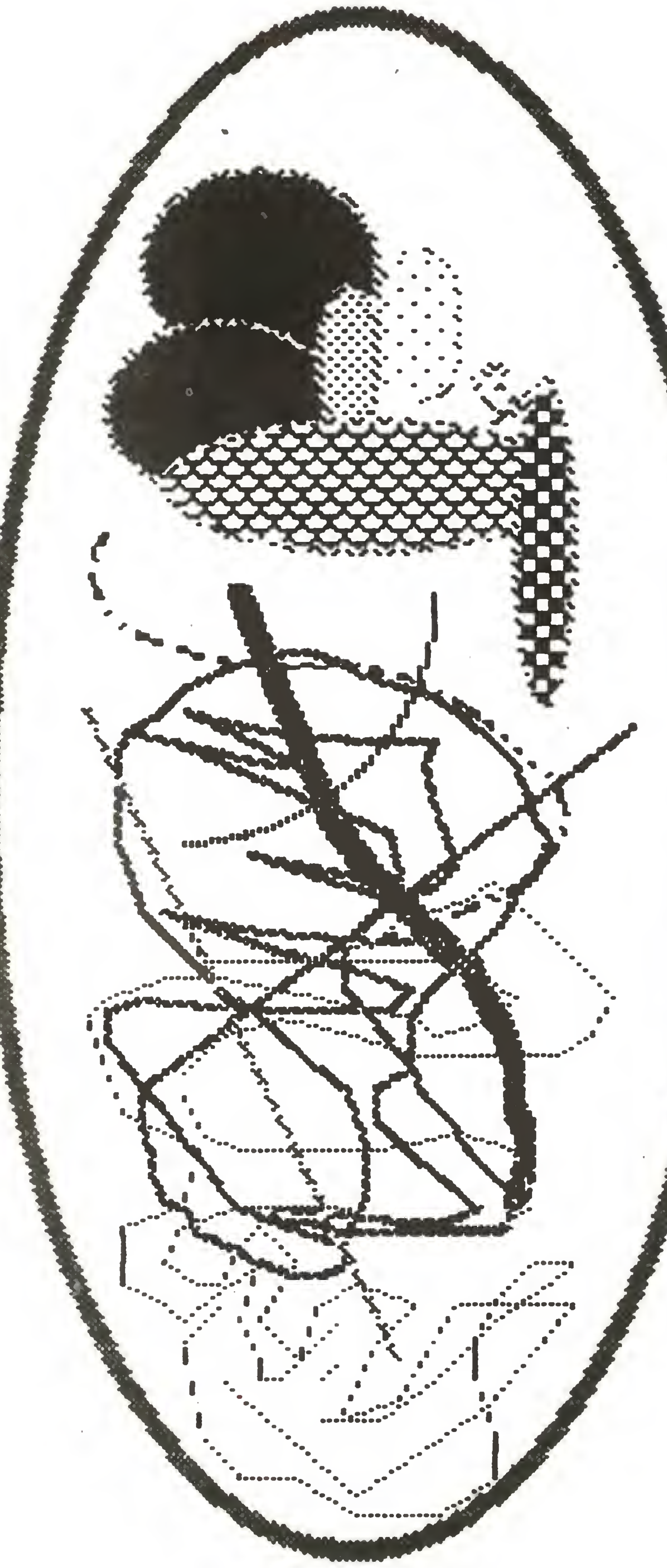
Żaden problem jeśli nie podoba się nam krój czcionki, mamy pod ręką całe kaszty (no, raczej dyski!):

abcdefgABCDEFG

abcdefgABCDEFG

abcdefgABCDEFG

abcdefgABCDEFG



Jeżeli ktoś czuje w sobie powołanie, może sam zaprojektować krój czcionki, zrobić rysunki, nawet włączyć zdjęcia w tekst. Po skończonej pracy przesyłamy gotowy wytwór naszej fantazji do właściwego punktu, zajmującego się drukowaniem na drukarce laserowej (rozdzielczość 4 razy lepsza niż w tym tekście, niestety koszt 4,5 tys$...), poczem robimy matryce i ... kręcimy wałkiem! Szkoda, że nie u nas. A może jednak dożyjemy tego szczęśliwego dnia?!

Napisał, złożył oraz wydrukował przy pomocy komputera Macintosh

Jakub Tatarkiewicz

Chińszczyzna i udawanie Greka

Komputerowe ludzie ładnie mówić po angielski, a kto nie mówić – tego bęc!

Producenci komputerów, początkowo wyłącznie anglosascy, po macoszemu traktowali liczne narody posługujące się językami innymi niż angielski i alfabetami innymi niż łaciński. Dominacja języka angielskiego początkowo nie budziła większych sprzeciwów, przy komputerach stawali bowiem do pracy naukowcy i inżynierowie mający do czynienia z najnowocześniejszą aktualnie techniką, zatem najczęściej znający angielski.

Udawanie Greka nie przynosiło większej szkody kasie producentów, dopóki mikrokomputery nie trafiły pod strzechy. Rozpowszechnienie się tych urządzeń, czemu sprzyjały coraz niższe ceny, wywołało popyt na oprogramowanie przyjazne wobec użytkownika. Potem kryzys na rynku mikrokomputerów, trwający już od półtora roku (dopiero teraz pojawiły się pierwsze oznaki, że może rychło się skończy) skłonił wytwórców do zauważenia, że Chińczycy, Arabowie, Hindusi e tutti quanti stanowią znaczną część ludzkości, której można sprzedać więcej komputerów. Zbyt jest potencjalnie przeogromny, ale pod warunkiem, że rozszerzy się ofertę oprogramowania o różne języki i pisma.

Japończycy, jako przodujący w produkcji mikro- i superkomputerów, sami się zatroszczyli o siebie. Już są bardzo blisko rozwiązania problemu akceptowania przez komputery – a ściślej przez urządzenia do przetwarzania tekstów – równoczesnego stosowania trzech systemów pisma*).

W Ameryce najpierw zabrano się do komputeryzowania chińszczyzny, gdyż jeśli nawet nie poziom zamożności, to przynajmniej sama liczba ludności pozwalała rokować nadzieje na ogromny zbyt prostych urządzeń. Pozwoliłoby to na opłacalny zwrot, w rozsądnym czasie, ogromnych nakładów na oprogramowanie.

IBM (oraz kilka firm tajwańskich i chińskich) próbowało pismo komputeryzować tym sposobem, że każdemu piktogramowi przypisywano piętrowy kod liczbowy. Piszący musiał rozbierać w swym umyśle znak na poszczególne kreski i wprowadzać z klasycznej klawiatury przypisane im kody. Pisanie było w całym tego słowa znaczeniu kodowaniem. Programy były zbudowane "technicznie", pomijano sposób funkcjonowania pisma w kulturze i nawyki w posługiwaniu się nim.

Amerykańska firma Asiagraphic Corp. z Port Jefferson (stan Nowy Jork) wymyśliła coś, co wprawdzie nie rewolucjonizowało całej sprawy, ale stanowiło istotny krok naprzód. W jej oprogramowaniu słowo chińskie należy wprowadzić do komputera w transkrypcji fonetycznej pinyin przyjętej w ChRL. Potem wystarczy wcisnąć klawisz opisujący tonację, w jakiej wymawia się dane słowo, potem jeszcze tylko symbol wskazujący, do której z 52 "podstawowych rodzin" należy dany znak – i już po kłopocie. Na ekranie pojawia się znaczek.

Strawniejsze jakby oprogramowanie do pisania piktogramów proponuje inna amerykańska firma, Intech Systems z Minneapolis. Mianowicie komputer nie ma klawiatury, a wyboru znaku dokonuje się przez dotknięcie ekranu palcem. Na dole ekranu stale wyświetla się litery łacińskie i dotykając liter tworzących sylabę – na ekran wywołuje się chmurę znaków oraz symbole odpowiadające różnym tonacjom wymowy. Trzeba więc znowu wskazać palcem. Potem już tylko wciśnięcie klawisza odsyłającego znak na samą górę ekranu, do wiersza lub słupka (można pisać i tak, i tak, zgodnie z chińskimi obyczajami) i można przejść do innego słowa. Europocentryzm przejawił się tutaj w tym, że punktem wyjścia jest transkrypcja na alfabet łaciński. Niemniej jednak komputer ma w pamięci 13 000 piktogramów, co wystarcza osobie posługującej się bardzo szerokim zasobem słów. Obok pinyin można wykorzystać transkrypcję używaną na Tajwanie – Chinese National Phonetic Alphabet. Drukarka mozaikowa o dużej gęstości umożliwia wydrukowanie piktogramów wcale wiernie, nawet z przeplataniem alfabetu łacińskiego i liczb arabskich. Komputer uzupełnia liczydło – liczy się na ekranie, dotykając palcem – oraz edytor tekstów.

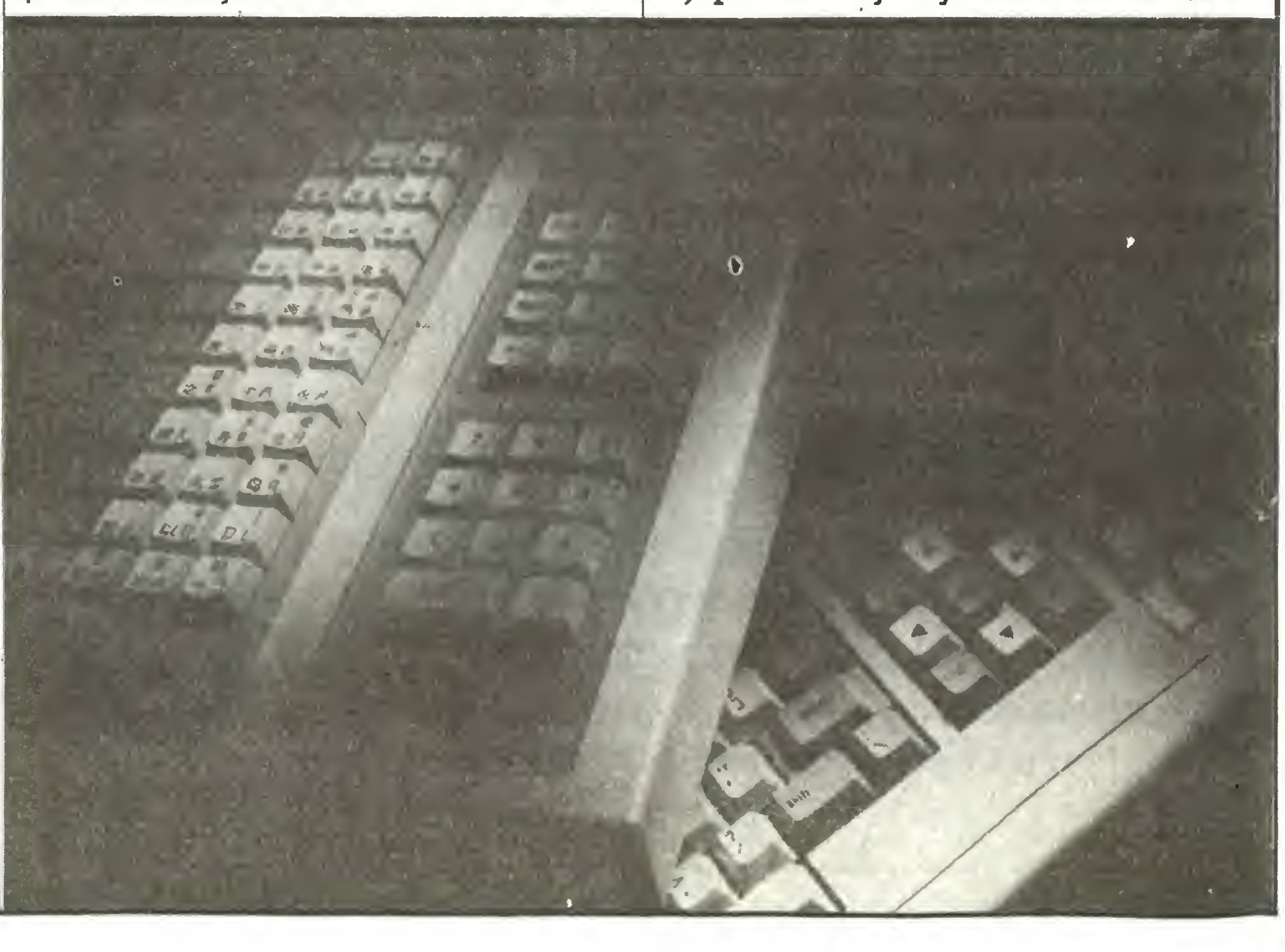
Firma Intex Software Systems International z Nowego Jorku ma gotowe oprogramowanie dla edytorów tekstu dla 11 języków z pismem alfabetycznym, w tym hebrajskiego, arabskiego, rosyjskiego i urdu (urzędowy w Pakistanie). Do końca 1986 r. kolejnych 27 grup użytkowników uszczęśliwi się oprogramowaniem w ich własnym języku i piśmie. Tutaj działa to tak: na jednym dysku elastycznym mieści się do 8 języków; po wybraniu jednego z nich klawiatura automatycznie przestawia się na dany język lub pismo. Trzeba tylko korzystać z pomocniczej kartki informującej gdzie jest co, bowiem klawiatura nie jest skażona żadnymi napisami; za to instrukcje pojawiają się na ekranie w wybranym języku. Pisać można w lewo (np. arabski, hebrajski) lub z lewa na prawo – jak my to czynimy. Można na jednej kartce papieru wydrukować coś po angielsku i arabsku – na przykład korespondencję oficjalną z Libijską Arabską Dżamahariją Ludowo-Socjalistyczną, która nawet z zagranicy nie przyjmuje tekstów bez arabskiego.

Prof. Władysław M. Turski opowiada, jak to na Zachodzie nabył drukarkę znającą 16 alfabetów, w tym czeski, estoński i dwa duńskie. Ale urządzenia z naszymi "ąęóźż" oraz "śńć" – nigdzie nie widział.

Mozemy wiec poki co poslugiwac sie wlasnym jezykiem w calej rozciaglosci – ale nie na komputerach.

JACEK A. LIKOWSKI

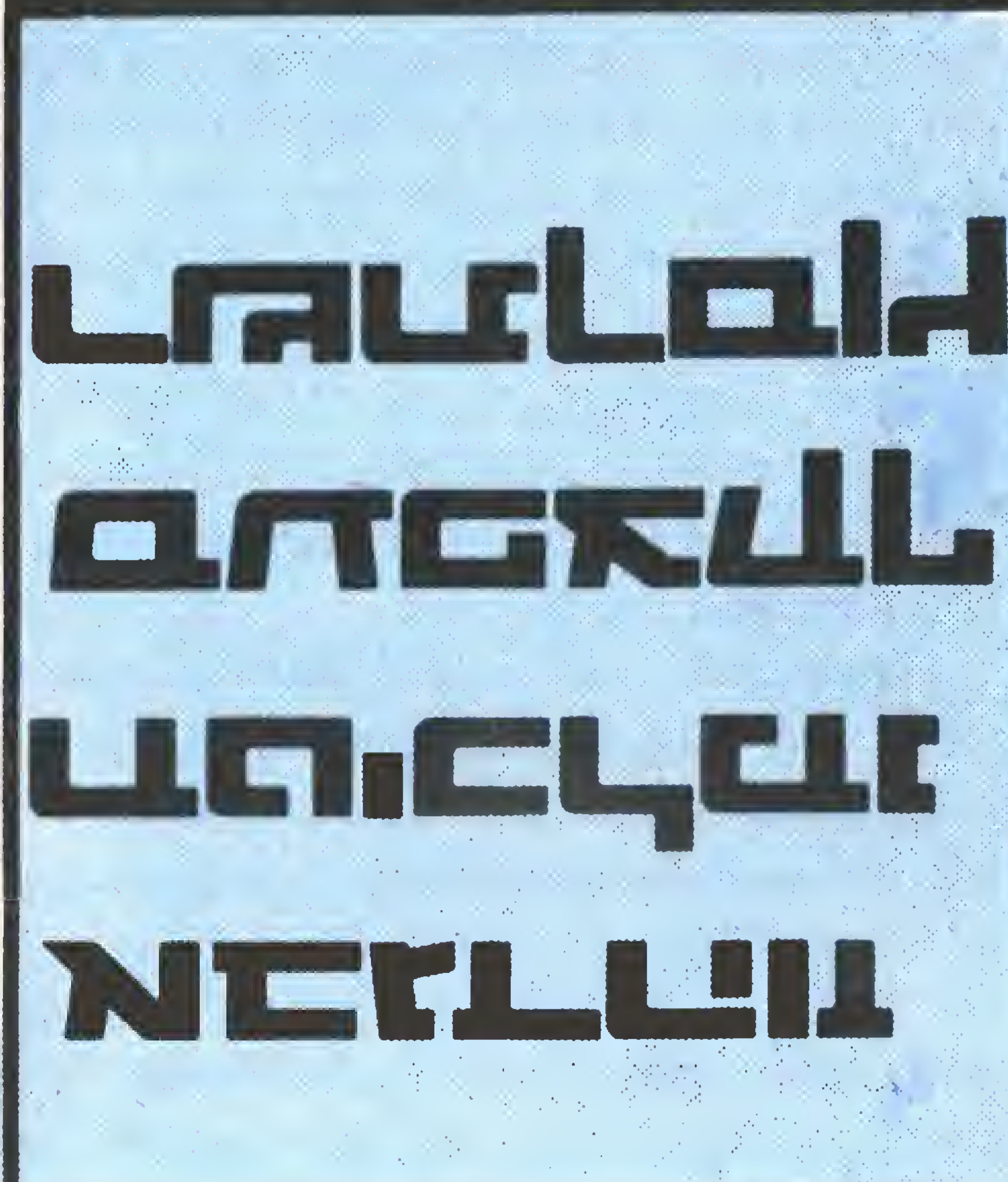
*) patrz dalej artykuł na ten temat.



AaBbCcDdEeFfGgHhIiJjKkLlŁłMmNnOoPpQqRrSsTtUuVvWwXxYyZz1234567890.?!&$£[]



Czytniki pisma hebrajskiego



Pismo hebrajskie nie sprawia większych kłopotów programistom. Poza tym, że pisze się z prawa w lewo, czyli odwrotnie niż w kulturze łacińskiej, jest to system alfabetyczny, same zaś znaki są bardzo proste. Niemniej jednak do niedawna nie było do niego czytników i oprogramowania, choćby takich, jakie są do pisma łacińskiego.

Tymczasem jest ogromna potrzeba takich czytników. Wprowadzenie ich postuluje mieszczący się w Nowym Jorku Komitet Ochrony Sztuki Rękopiśmiennej, grupujący kaligrafów hebrajskich, którym chodzi o komputerową dokładność przy weryfikacji tekstów. Niektórzy wprawdzie uważają, że nic nie jest w stanie zastąpić skryby, który przystępuje do mozolnego zadania wyszukania błędów w odpowiednim stanie ducha uzyskanym dzięki modlitwom. Jednak wspomniany Komitet zamierza w br. uruchomić 14 przenośnych punktów weryfikacji tekstów religijnych, które będą działały w USA, Europie Zachodniej i Izraelu.

Religijne pisma hebrajskie muszą być pisane ręcznie i nie mogą mieć żadnych wad, za które uważa się opuszczenia linijek, niewłaściwy kształt liter lub nawet ich stykanie się ze sobą. Trud weryfikacji tekstów religijnych, w tym *Tor*, które odczytuje się głośno w synagogach, jest ogromny. Nic zatem dziwnego, że inicjatywa stworzenia odpowiedniego oprogramowania wyszła z kręgów niezwykle tradycyjnych – ludzi, których zawodem jest artystyczna kaligrafia, kopiowanie i weryfikacja tekstów. Ale mimo komputeryzacji kaligrafom (których w USA jest około 200) i tak pozostanie wiele przy poprawianiu błędów.

(JAL)

Komputerowe przetwarzanie tekstów po japońsku

Gdy słyszymy, że na świecie, a zwłaszcza w angielskim obszarze językowym, narzędziem powszechnego użytku stają się programy i urządzenia komputerowe do przetwarzania tekstów, umiejące nie tylko samoczynnie kontrolować poprawnośc ortograficzną, gramatyczną i językową wprowadzanego tekstu, ale także automatycznie rozmieszczać go na ekranie – ogarnia nas zrozumiała zazdrość i żal. Zdaniem wielu najwybitniejszych, również polskich informatyków, opracowanie podobnych programów dla języka polskiego byłoby zadaniem tak trudnym, że wręcz niewykonalnym. Przeszkodą nie do przezwyciężenia jest ponoć nasza fleksja i odmiana czasowników.

Dla wielu ludzi na świecie język chiński jest synonimem niezrozumiałości, trudności i komplikacji. Trudno więc oprzeć się zdziwieniu na wieść, że Japończycy, posługujący się językiem bardzo podobnym do chińskiego, zdołali poradzić sobie z problemem budowy procesora tekstu i że jest on... szybszy i łatwiejszy w obsłudze niż jego angielskojęzyczne odpowiedniki.

Przetwarzanie tekstów staje się obecnie w Japonii, podobnie jak w USA i Europie Zachodniej, jednym z głównych zastosowań komputerów i każdy producent, który myśli o sukcesach na tym polu, musi wyposażać swój sprzęt w oprogramowanie pozwalające porozumieć się z komputerem po japońsku.

Mimo podobnych funkcji, procesory tekstu japońskiego muszą poradzić sobie z zupełnie innymi problemami niż procesory angielskojęzyczne. W języku japońskim nie oznacza się bowiem przerw między słowami, lecz pisze się je w nieprzerwanym ciągu, stawiając znaki jeden za drugim. Uzyskanie tekstu w postaci szpalt gotowych do druku nie jest więc żadnym problemem. Istota trudności leży natomiast w tym, że stosuje się tam równolegle trzy rodzaje ortografii: systemy kandżi, hiragana i katakana.

Pierwszy z nich to zestaw znaków ideograficznych pochodzenia chińskiego, posiadających samodzjelne znaczenie, dwa pozostałe zaś składają się z fonetycznego zapisu sylab języka japońskiego. Liczba (zasób) słów systemu kandżi wynosi ponad 10 tys., z czego 1850 opanowują uczniowie w ramach obowiązkowego, dziewięcioletniego systemu kształcenia podstawowego. Systemy hiragana i katakana składają się z 50 znaków podstawowych.

W tekstach japońskich wszystkie trzy systemy są stosowane z identyczną "częstotliwością". Słowa posiadające integralne znaczenie (czasowniki, rzeczowniki, przymiotniki etc.) zapisywane są "w kandżi", natomiast ich końcówki – są w systemie hiragana. Do fonetycznego zapisu słów obcego pochodzenia używa się systemu katakana.

Wprawdzie każdy japoński tekst może być zapisany 50 znakami zestawu hiragana, tj. 50 sylabami podstawowymi, ale w systemie kandżi myśli mogą być przedstawione w sposób bardziej zrozumiały i zwięzły. Do zrozumienia tego zapisu trzeba jednak znać... historię kultury japońskiej, a często i chińskiej.

I tak np. znak kandżi 日 przed tysiącami lat miał formę koła, dopiero później stał się "kwadratowy" Można więc zrozumieć dlaczego znak ten oznacza *słońce*. Kandżi 木 oznacza *drzewo*, kandżi 林 oznacza *park*, zaś kandżi 森 oznacza *las*.

Jeśli podkreślimy dolną część kandżi drzewo, to powstanie 本 o znaczeniu *korzeń* lub *geneza*, *pochodzenie*. Z kombinacji drzewa i słońca powstanie nowe kandżi 東. Oznacza ono *wschód*, bo przecież właśnie na wschodzie obserwujemy to zjawisko: słońce wschodzi zza drzewa.

Jeśli teraz kandżi 日 i 本 połączymy w jeden znak, to powstanie kombinacja znaków 日本 oznaczająca *Japonię*. Japonia jest bowiem "Krajem Słonecznego Pochodzenia"; legendy mówią, że pierwszy cesarz, Amaterasu Omikamina, był wnukiem Bogini Słońca.

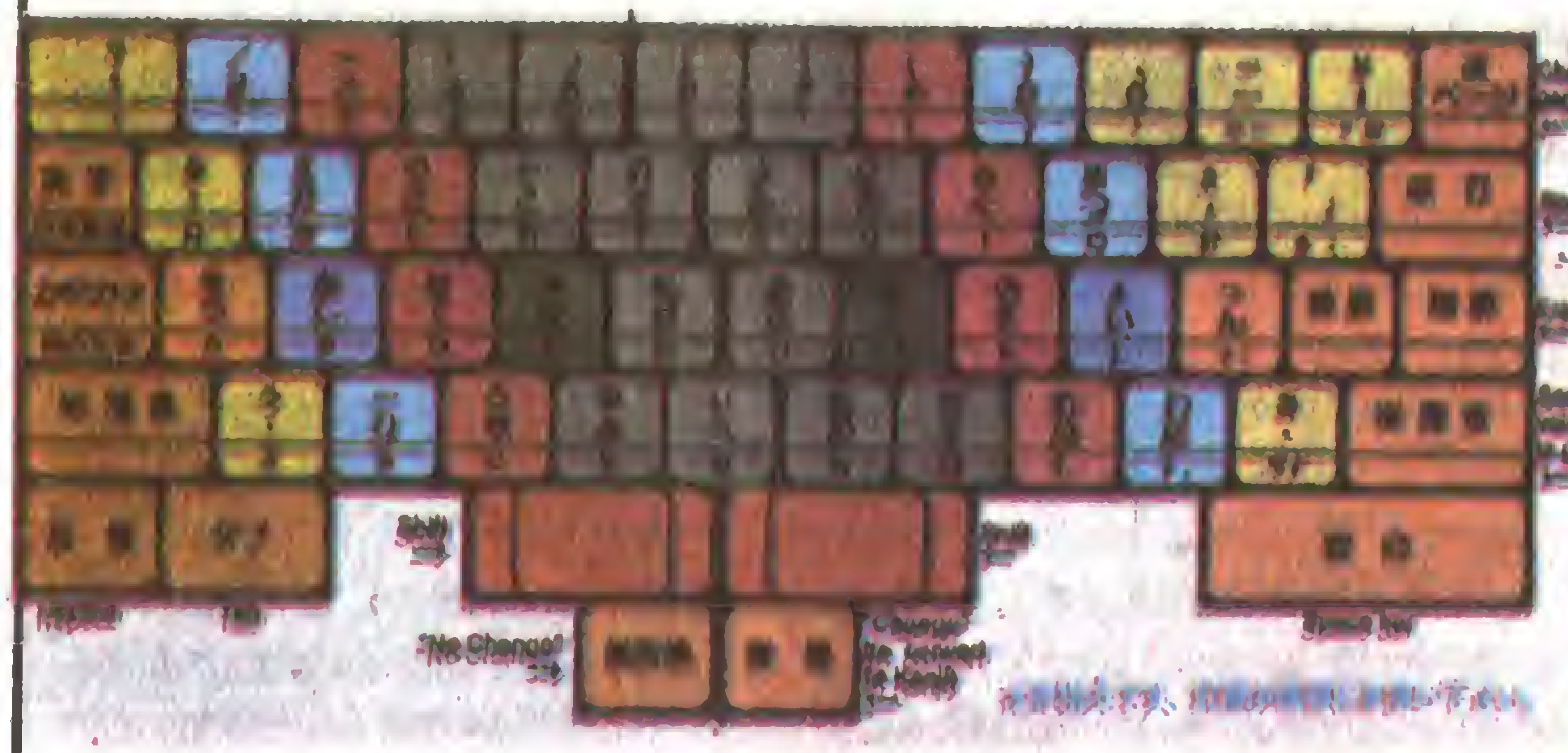
Kandżi 京 oznacza *miasto*, ewentualnie *stolicę*. W tym układzie jasne jest, że 東京 oznacza *Tokio*, dosłownie zaś *Wschodnią Stolicę*, ponieważ spośród trzech stolic Japonii, Tokio leży najdalej na wschód.

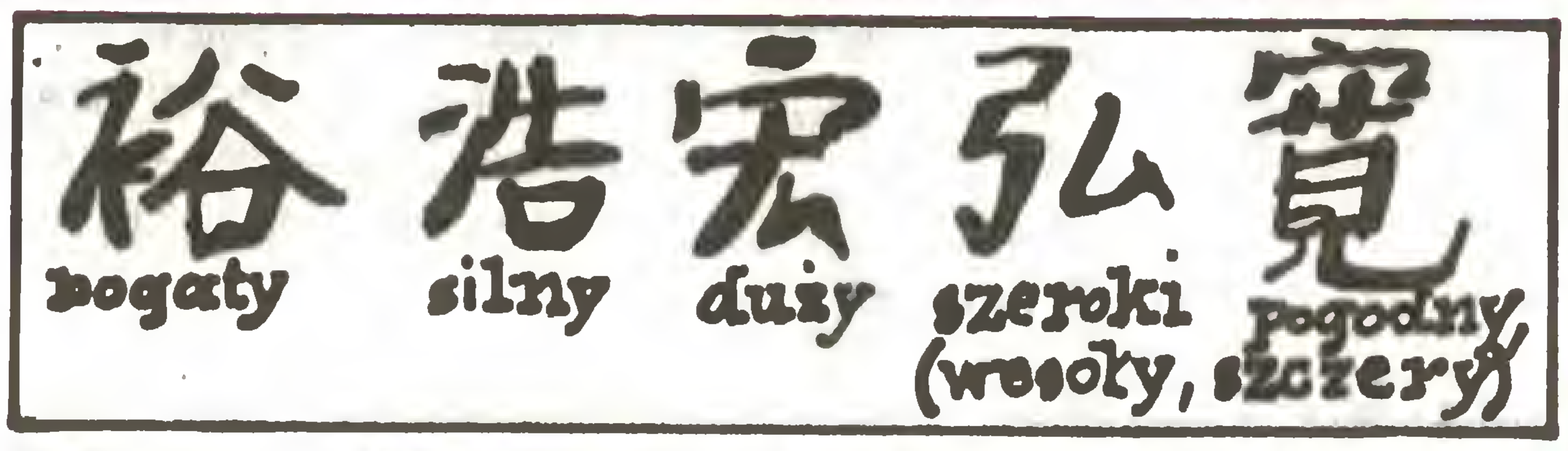
Jeśli już znamy znak ideograficzny oznaczający wschód, zobaczmy jak wyglądają znaki dwu innych kierunków. 北 oznacza *północ* natomiast 南 oznacza *południe*. Ponieważ w Chinach używa się tego samego systemu zapisu, to z łatwością zrozumiemy, że zapis Pekinu 北京 oznacza *Północną Stolicę* zaś Nankinu 南京 *Południową Stolicę*.

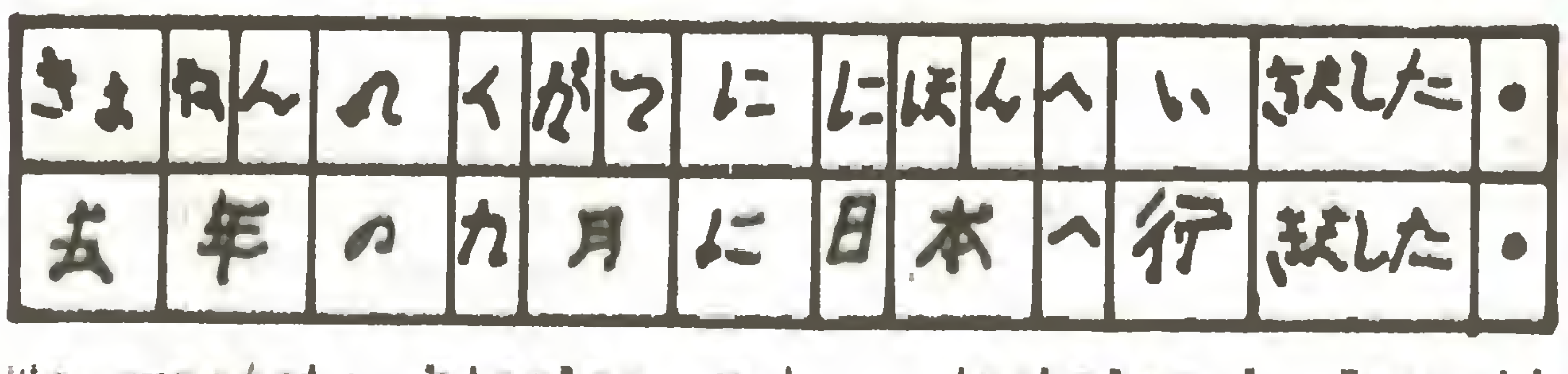
Poza przytoczonymi zaletami, które wiążą się z licznymi ułatwieniami w zrozumieniu, system zapisu kandżi ma także inne walory. Jedną z cech języka japońskiego jest to, że składają się nań liczne słowa o takiej samej wymowie, lecz o odmiennym znaczeniu. Te różnice znaczeniowe mogą być ukazane jedynie w systemie zapisu kandżi.

Jedno z najczęściej spotykanych japońskich imion Hiroszi może mieć np. aż pięć znaczeń w zależności od tego, jakie kandżi będzie użyte do zapisania tego słowa (por. rys 1).

Jeśli to słowo zapisalibyśmy w transkrypcji na litery alfabetu łacińskiego, albo jednym z dwóch zapisów sylabowych (np. hiraganą w sposób następujący ひ (wymowa hi) ろ (wymowa ro) し (wymowa szi), to wówczas różnice znaczeniowe, które są obecne w systemie kandżi, stałyby się nieuchwytne.







We wrześniu ubiegłego roku pojechałem do Japonii.

Jasne jest więc, że zapis ideograficzny ma wiele zalet w porównaniu do zapisu składającego się z liter. Wadą systemu kandżi jest to, że opanowanie wielu tysięcy znaków zajmuje ok. 10-15 lat, pisanie zaś na maszynie jest czynnością uciążliwą i trwa bardzo długo.

Na klawiaturze komputera wystukuje się tekst tylko w fonetycznym zapisie zgłoskowym (a więc korzystając z hiragany lub katakany składających się tylko z 50 znaków), po czym urządzenie – spośród wielu tysięcy znaków – wybiera kandżi, które odpowiadają poszczególnym znakom. Ukazują się one następnie na ekranie, co sprawia, że można z łatwością wybierać odpowiedni znak, który następnie trafia do ostatecznej wersji tekstu.

Brzmi to nieprawdopodobnie, a jednak: przy zastosowaniu tego rodzaju procesora i posłużeniu się znakami trzech systemów zapisu można uzyskać tekst gotowy do druku szybciej, niż byłby pisany według zasad fonetyki angielskiej przy użyciu maszyny do pisania z łacińską czcionką. A oto konkretny przykład. Przypomnijmy sobie "przypadek" z imieniem Hiroszi. Aby napisać to imię, na klawiaturze trzeba nacisnąć 9 klawiszy: jeden ze względu na podniesienie wałka, ponieważ chodzi o dużą literę, siedem liter i wreszcie odstęp międzysłowny. Natomiast w przypadku zastosowania wspomnianego procesora – tylko pięć klawiszy: trzy klawisze sylabowe ひ ろ し klawisz charakteru zapisu i wreszcie ostatni klawisz, który spośród znaków ukazujących się na monitorze wybiera odpowiednie kandżi (znaki ideograficzne).

Na zakończenie przedstawię jedno zdanie pisane znakami hiragana, tak jak jest ono wprowadzane do wordprocesora. Powyżej podaję to samo zdanie w zapisie kandżi. Zdanie brzmi: we wrześniu ubiegłego roku pojechałem do Japonii. (rys. 2) Warto przy tym zauważyć, że do napisania tego zdania potrzeba tylko 27 uderzeń, jeśli dysponujemy prezentowanym procesorem tekstu. Polska maszyna do pisania wymaga zaś 53 uderzeń.

Opracował: SZCZEPAN WORONOWICZ

Wg: Gábor Márkus, Word processing Japánban, (Word processing w Japonii), Mikroszámitógép Magazin, nr 3/1985, s. 25 – BWE

KLAWIATURA DVORAKA

Pierwsza maszyna do pisania pochodzi z epoki, gdy powszechnie używano gęsich piór i jest aż o sto lat starsza od... stalówki. Opatentował ją w Anglii Henryk Mill w roku 1714.

Jednak maszyną do pisania, od której wywodzą się wszystkie współczesne, jest dopiero konstrukcja Amerykanina Ch. L. Sholesa z roku 1872, którą w rok później zakupił przemysłowiec Remington, rozpoczynając masową produkcję. Sholes jest też autorem układu klawiatury, w której rząd zawierający litery zaczyna się od Q, a następne czcionki to W, E, R, T, Y. Stąd nazwa QWERTY dla całego układu niemal niezmienionego do dziś we wszystkich krajach posługujących się alfabetem łacińskim. Niemal – bo np. w Polsce układ nie odpowiada już ściśle nazwie, jako że szóstą czcionką w rzędzie liter jest nie Y lecz Z. Oczywiście trzeba było też dokonać kilku innych przeróbek, by dostosować klawiaturę do polskich znaków.

Czym kierował się Sholes rozmieszczając właśnie w taki sposób poszczególne symbole? W dzisiejszej dobie standaryzacji i dążenia do maksymalnej efektywności pytanie może wydawać się banalne. Tymczasem, wbrew temu co zapewne sądzi większość czytelników, układ QWERTY ma na celu bynajmniej nie ułatwienie szybkiego pisania, lecz wręcz przeciwnie – uniemożliwienie tego... Nie chodziło przy tym oczywiście o żaden sabotaż na ogólnoświatową skalę. Po prostu w ówczesnych maszynach do pisania czcionki miały tendencję do zakleszczania się. Wymuszenie mniejszej efektywności było więc koniecznością.

W początku lat trzydziestych naszego stulecia August Dvorak, profesor statystyki Uniwersytetu stanu Washington w Seattle, zaproponował zastosowanie całkiem odmiennego układu klawiatury. W odróżnieniu od swego poprzednika kierował się dążeniem do uzyskania możliwie dużej prędkości pisania i zminimalizowania niezbędnego wysiłku piszących.

Zdaniem Dvoraka, nowa klawiatura pozwala na zwiększenie szybkości pisania, z jednoczesnym zmniejszeniem liczby błędów, od około 35 do 50 procent. Z prowadzonych przez niego badań wynikało również, że radykalnie zmniejsza się wysiłek piszących. Liczył drogę, jaką muszą przebywać palce maszynistki przy zastosowaniu obu typów klawiatur. Twierdził, że użycie jego układu zmniejsza tę drogę od 12 do 20 razy! Faktem jest, że uczniowie Dvoraka regularnie wygrywali zawody na szybkość pisania w latach trzydziestych i czterdziestych. Również najszybsza, według księgi Guinness'a, maszynistka świata, Barbara Blackburn, używa klawiatury Dvoraka.

Mimo tych sukcesów nowy system z trudem toruje sobie drogę do osiągnięcia pozycji nowego standardu. Zainteresowanie wzrosło dopiero w ostatniej dekadzie w związku z wprowadzeniem elektronicznych klawiatur. Są już dostępne oddzielne klawiatury z układem Dvoraka do komputerów IMB PC. Najdalej poszła firma Apple. W swoim modelu Apple IIc wbudowali przełącznik, dzięki któremu ta sama klawiatura może pracować w obu systemach. Dla komputerów IBM PC, Apple IIe, niektórych modeli Tandy i Commodore 64 dostępne są programy pozwalające interpretować klawisze zgodnie z nowym układem. Istnieje również szereg programów ułatwiających naukę pisania na zmodyfikowanej klawiaturze.

Ale czy klawiatura Dvoraka jest naprawdę tak dobra, żeby warto było uczyć się maszynopisania od nowa? Przeprowadzone niedawno niezależne badania wykazały rzeczywistą wyższość układu Dvoraka nad QWERTY, jednak nie aż tak wielką, jak sugerował to jego autor. Zdaniem grupy badaczy z Uniwersytetu stanu California w San Diego, klawiatura Dvoraka pozwala zwiększyć prędkość pisania zaledwie o 5 do 10 procent. Wyznaczanie drogi przebywanej przeciętnie przez palce maszynistki wypadło na korzyść nowego układu w stosunku 1:1.5, a nie co najmniej 1:12, jak utrzymywał sam Dvorak. Wyniki tych prac zostały opublikowane przez Donalda R. Gentnera i Donalda A. Normana w artykule pt. "The Typist's Touch" w Psychology Today w marcu 1984 roku. Podobne wyniki uzyskali później również i inni badacze.

Przed przystąpieniem do przeróbki własnej maszyny do pisania lub klawiatury komputera warto sobie uzmysłowić, że wszystkie te badania były prowadzone pod kątem potrzeb języka angielskiego. Optymalny układ znaków dla polskiej maszyny może okazać się zupełnie inny.

Opracował: A. KADLOF

EDYTOR TEKSTU "TEKST/ED"

Program "TEKST/ED" – opracowany na podstawie programu "TASWORD TWO" firmy TASMAN SOFTWARE – służy do redagowania tekstów za pomocą ZX Spectrum. Do typowych zastosowań programu należą: pisanie artykułów, referatów, pism urzędowych, tłumaczeń itp. Tekst wprowadzamy z klawiatury do pamięci komputera obserwując na ekranie, czy nie ma w nim pomyłek. Gotowy tekst przeglądamy na ekranie, a po wprowadzeniu ewentualnych poprawek oraz sprawdzeniu prawidłowego układu tekstu (strony, tytuły, wcięcia itp.) żądamy wydruku na drukarce lub elektrycznej maszynie do pisania. Tekst zapisujemy również na kasecie magnetofonowej lub dyskietce – może on przydać się w przyszłości do sporządzenia nowej jego wersji lub też jako część większej całości.

Zalety takiego redagowania tekstu w porównaniu z tradycyjnym pisaniem na maszynie są następujące:

- Przy wprowadzaniu zmian (korekty błędów, przeredagowanie zdań i akapitów, przeniesienie fragmentów w inne miejsce, inne rozmieszczenie tytułów itp.) nie przepisuje się całego tekstu od nowa – wystarczy podać odpowiednie rozkazy oraz wpisać w określone miejsca nowe wersje tekstu.
- Ze zgromadzonych tekstów można tworzyć różne kombinacje: np. przed wydrukowaniem zmieniając tę część, która dotyczy konkretnego adresata.
- Wydruk kolejnej wersji tekstu uzyskuje się bardzo szybko – jest to niezmiernie istotne przy projektowaniu wyglądu zewnętrznego poszczególnych arkuszy; dodatkowe możliwości oferują tu drukarki mozaikowe (różne kroje czcionek, druk wytłuszczony, rozszerzony i in.).
- Program przejmuje niektóre czynności redakcyjne, uwalniając od nich piszącego – przede wszystkim dotyczy to decyzji o przejściu do nowego wiersza, wyrównywania lewego i prawego marginesu oraz uporządkowania akapitu po wprowadzeniu zmian.

Program "TEKST/ED" rozszerza możliwości ZX Spectrum w trzech dziedzinach istotnych dla przetwarzania tekstów:

- na ekranie widoczne są 64 znaki w wierszu zamiast 32,
- wszystkie polskie litery (ą, ó, ę itd.) dostępne są na klawiaturze, ekranie i drukarce,
- obsługa różnych drukarek (D1, SEIKOSHA GP5, GP1, GP5, DZM18) i interfejsów (PB3, KEMPSTON, ZX INTERFACE 1) wbudowana jest w program.

KORZYSTANIE Z PROGRAMU

Podczas wprowadzania tekstu komputer zachowuje się tak, jak maszyna do pisania – wciśnięcie klawisza powoduje przyjęcie znaku do tekstu z wypisaniem go na ekranie i przesunięciem kursora. Kursor jest to migający czarny kwadracik – pokazuje on, w które miejsce będzie wpisany następny znak. Kursorem można dowolnie "jeździć" po ekranie w celu dopisania lub poprawienia tekstu w określonych miejscach. Obraz na ekranie jest odzwierciedleniem tego, co później "wyjdzie" z drukarki (nie uwzględnia on jedynie różnych krojów pisma, natomiast widoczne są miejsca, w których będą one zmieniane). Ekran stanowi jak gdyby okno, przez które widzimy fragment tekstu (22 wiersze), a które możemy ustawić w dowolnym miejscu. Jest ono również przesuwane automatycznie w miarę wprowadzania tekstu.

Niektóre klawisze lub ich kombinacje mają znaczenie szczególne – zamiast wpisania znaku wywołują określone funkcje programu. Przykładem tego są klawisze ze strzałkami – zmieniają one pozycję kursora (dotyczy to klawiatury Spectrum – na klawiaturze Spectrum Plus należy przełączyć górny rząd klawiatury na cyfry – klawiszem "1"). Specjalną rolę pełni klawisz "q". Ponieważ w polskich tekstach litera ta występuje dość rzadko, jest on przeznaczony do wywoływania różnych funkcji, stając się dodatkowym klawiszem "SHIFT" (literę "q" uzyskuje się poprzez kombinację "SYMBOL SHIFT-q"). Między innymi przy jego pomocy wpisuje się polskie litery – wciskając równocześnie "q" i jeden z klawiszy: "a", "c", "e", "l", "n", "o", "s", "z", "x" wpisujemy odpowiednio literę: ą, ć, ę, ł, ń, ó, ś, ż, ź. Ten sam klawisz w połączeniu z inną literą powoduje wywołanie jednej z funkcji: ustawienie marginesu, uporządkowanie akapitu, zaznaczenie, przeniesienie lub przekopiowanie bloku tekstu, ustawienie tytułu na środku wiersza, wyszukanie żądanego słowa i ew. zastąpienie innym, skasowanie tekstu i in. Kombinacje "qd" i "qm" służą do współpracy z drukarką i magnetofonem (lub inną pamięcią zewnętrzną – ZX MICRODRIVE, floppy itp.) – na ekranie pokazywana jest w formie "MENU" lista możliwych operacji. Natomiast naciskając klawisze "qi" użytkownik otrzymuje na ekranie listę funkcji programu z podaniem klawiszy.

Na ekranie poza tekstem widoczne są pewne informacje pochodzące z programu: numer aktualnego wiersza i znaku, konfiguracja klawiatury, ustawienie przełączników. Klawiaturę konfiguruje się klawiszami "1", "2", "3", "4" i "9", ustawiając dolne 3 rzędy na duże lub małe litery, a górny rząd na: cyfry, kursory, operacje na wierszach (liniach) tekstu, przeglądanie (rolowanie) tekstu lub wprowadzanie znaków graficznych (służą one jako znaki sterujące drukarką – zmiana kroju pisma, przejście do następnej strony itp.). Istnieje możliwość wyboru, który zestaw będzie dostępny bezpośrednio, a który poprzez "CAPSSHIFT".

Przełączniki obrazują aktualnie obowiązujący tryb pracy programu (wstawianie znaków w środek tekstu, równanie do prawego marginesu, dzielenie słów między 2 wiersze, 32 znaki w wierszu), do ich zmiany służą klawisze "q1" do "q4".

Jeżeli dzielenie słów jest zabronione (przełącznik wyłączony), to słowo, które nie mieści się w danym wierszu, zostanie w całości przeniesione do wiersza następnego, a poprzedni wiersz zostanie (jeżeli włączone jest równanie do prawego marginesu) dosunięty do prawej strony poprzez wpisanie dodatkowych odstępów. Operacje te następują szybko i nie wymagają przerwy w pisaniu. Założenie jest takie, że pisać można bez patrzenia na ekran – program sam podzieli tekst na wiersze, a jeżeli dane słowo akurat mieści się na końcu wiersza, to program dosunie następne słowo do lewego marginesu kolejnego wiersza.

POPRAWIANIE TEKSTU

Skasowanie znaku lub jego zamiana na inny należą do czynności bardzo prostych – po ustawieniu kursora na niepotrzebnym znaku kasujemy go klawiszem "DELETE" ("" lub "CAPSSHIFT-" – w zależności do konfiguracji klawiatury) lub "nadpisujemy" innym znakiem. Klawisz "DELETE" przesuwa dodatkowo resztę wiersza w lewo (wyrównanie marginesu można zawsze przywrócić klawiszami "qu"). Natomiast wstawienie tekstu w środek wiersza jest nieco trudniejsze i wymaga zrozumienia procesu porządkowania akapitu. Jest to funkcja programu wywoływana klawiszami "qu" i polega na rozmieszczeniu słów w wierszach akapitu według ustawionych przełączników (równania marginesu i dzielenia słów). Jeżeli więc w wyniku skasowania kilku słów (poprzez zastąpienie ich odstępami) powstanie w środku wiersza luka, to program porządkujący usiłuje ją wypełnić, dosuwając pozostały tekst w lewo i wciągając jakby kolejne słowa z następnego wiersza, jeżeli się mieszczą. Proces ten kontynuowany jest w następnych wierszach aż do końca akapitu – tzn. do wystąpienia pustego wiersza lub wcięcia tekstu (odsunięcia początku wiersza od marginesu). Przystępując do wstawiania tekstu do danego akapitu, musimy więc ustawić lewy margines na tę samą wartość, jaką miał przy pisaniu akapitu. Jeżeli tekst będzie odsunięty od marginesu, to program potraktuje każdy wiersz jako początek nowego akapitu. Jeżeli natomiast margines zachodzi na tekst, to program nie uwzględni początkowych znaków każdego wiersza.

Margines lewy ustala się klawiszami "qw" – przedtem należy umieścić kursor na pierwszym znaku za marginesem. Ustawienie marginesu służy do wcięcia całych fragmentów tekstu i nie jest równoznaczne z marginesem na papierze – drukując tekst deklarujemy dodatkowy margines dla wydruku. Podczas redagowania tekstu możemy więc wykorzystywać pełną szerokość wiersza (64 znaki), co po dodaniu marginesu da standardowy format pisma maszynowego. Manipulując marginesami można pisać tekst w dwóch szpaltach.

Po ustawieniu marginesu możemy przystąpić do dokonywania wstawek. Przesuwamy kursor w odpowiednie miejsce, włączamy tryb wstawiania znaków ("q1"), a następnie żądamy rozsunięcia tekstu klawiszem "ENTER". W tak powstałą lukę wpisujemy nowy znak. Następny można wpisać po powtórnym wciśnięciu "ENTER". Jeżeli zażądamy rozsunięcia tekstu ustawiając kursor między słowami, to program zrobi od razu więcej miejsca, przenosząc resztę do następnego wiersza – takie łamanie wierszy stosuje się np. w przypadku zmiany marginesów w gotowym tekście. Po wprowadzeniu poprawek należy uporządkować akapit ("qu") i włączyć tryb wstawiania ("q1").

URZĄDZENIA PERYFERYJNE

W zakresie współpracy z magnetofonem program oferuje następujące możliwości:

- nagranie tekstu na taśmę, z ewentualną weryfikacją,

- wczytanie tekstu z taśmy,
- dołączenie tekstu z taśmy do tekstu w pamięci komputera.

Zamiast magnetofonu można zastosować pamięć zewnętrzną o szybszym dostępie (ZX MICRODRIVE, dyski elastyczne itp.). Dla niektórych systemów dyskowych istnieją również specjalne wersje programu.

Możliwość wydruku tekstu na drukarce stanowi kluczowy element każdego programu do przetwarzania tekstów. Istotny problem stanowi tu różnorodność spotykanych w kraju drukarek i interfejsów. Szczególnie wydrukowanie polskich znaków wymaga odrębnych programów dla różnych typów drukarek. W programie "TEKST/ED" wbudowana jest obsługa najpopularniejszych w kraju drukarek (D1ØØ, GP5ØS, GP1ØØ, GP5ØØ, DZM18Ø) i interfejsów (PB3-CENTRONICS i LOGABAX, KEMPSTON, ZX INTERFACE 1). Ponadto można wybrać typ "drukarka znakowa" – w tym przypadku program dokona przy wydruku konwersji liter polskich na łacińskie (ą na a, ć na c itd.) Wybierając opcję "interfejs własny" nakazuje się wykorzystanie kanałowego programu ZX Spectrum (obsługa drukarki ZX PRINTER lub GP5ØS bądź też zainicjowanego wcześniej własnego interfejsu użytkownika). Ponadto w programie znajduje się miejsce na specjalizowany moduł obsługi drukarki i/lub interfejsu. Takie moduły istnieją już m.in. dla drukarek STAR SG1Ø, PANASONIC, EPSON, SHINWA, BROTHER i in. oraz interfejsów LPRINT, APINA, TOS i in. Moduły te oferują również następujące możliwości dodatkowe:

– wplatanie grafiki w tekst,
– definiowanie własnych znaków (np. liter greckich),
– wysłanie sekwencji kodów sterujących.

Kaseta z programem zawiera moduł "gp5ØØ+" rozszerzający możliwości współpracy z drukarką GP1ØØ lub GP5ØØ. Łączenie grafiki z tekstem polega na wplataniu w tekst poszczególnych fragmentów i obrazka, przy czym należy zadeklarować, który jego wycinek ma być w danym miejscu wydrukowany i w jakiej skali (możliwe jest dwukrotne powiększenie). Obrazek może być utworzony przy pomocy własnego programu lub jednego z gotowych programów graficznych (np. "MELBOURNE DRAW" lub "DLAN") i musi zawierać wszystkie rysunki, które mają znaleźć się w tekście. Rysunek można umieścić między akapitami tekstu, jak również obok tekstu (w sąsiedniej szpalcie).

Program pozwala na podanie dodatkowych parametrów dla wydruku tekstu (druk fragmentu tekstu – od linii do linii, lewy margines itp.) a także na zmniejszenie szerokości wydruku (np. do 32 znaków dla drukarki GP5ØS).

Znaki kierowane na drukarkę przechodzą przez kanał 4 systemu operacyjnego Spectrum i mogą być przesłane na inne urządzenie (np. ZX MICRODRIVE lub inny komputer – poprzez łącze RS lub ZX NETWORK).

Użytkownikom zainteresowanym przetwarzaniem tekstów na ZX Spectrum polecić należy następujący zestaw urządzeń peryferyjnych:

– drukarka STAR SG1Ø (posiada rozbudowane możliwości graficzne, pisze na papierze perforowanym i na pojedynczych arkuszach, wymaga zwykłej taśmy od maszyny do pisania) – z dowolnym interfejsem,
– dyski elastyczne (system TOS, SP/M lub BETA).

PAKIET PROGRAMÓW "POLONEZ"

Program "TEKST/ED" wchodzi w skład zestawu programów, którego celem jest umożliwienie polskiemu użytkownikowi wykonania podstawowych operacji z dziedziny przetwarzania danych przy pełnych możliwościach stosowania polskich liter. Odpowiednie wersje pakietu przystosowane są do obsługi konkretnych drukarek, interfejsów i ewentualnie jednostek dyskowych. W skład pakietu wchodzą następujące programy:

- "TEKST/ED" – wersja Ø2 (magnetofon/microdrive) lub Ø2D (magnetofon/floppy).
- "MFPOL" – adaptacja programu "MASTERFILE" (CAMPBELL) – baza danych z polskimi literami we wszystkich krojach pisma (32, 42 i 51 znaków w wierszu). Program uwzględnia kolejność leksykograficzną według polskiego alfabetu; wydruki na drukarce w formie graficznej kopii ekranu lub specjalnych raportów w formacie ustalonym przez użytkownika.
- "OMPOL" – adaptacja programu "OMNICALC 2" (MICROSPHERE) – program do obliczeń tabelarycznych (kalkulacje finansowe, modele ekonomiczne, zestawienia statystyczne itp.) z możliwością rysowania histogramów.
- "LITERY" – rozszerzenie języka BASIC o polskie litery na klawiaturze, ekranie i drukarce.

Ponadto w skład zestawu wchodzą pewne programy użyteczne dla danej konfiguracji sprzętu (np. kopiowanie ekranu na drukarkę). We wszystkich programach przyjęto jednakowe kody polskich liter i (poza "TEKST/ED") taką samą zasadę obsługi klawiatury (najpierw klawisz "q" a następnie "a", "c", "e", "l", "n", "o", "s", "z", lub "x".

TADEUSZ WILCZEK

Program malutki, malutki, ale ciekawiutki

Prezentowany program zamienia standardowy zestaw znaków (litery, cyfry, znaki przestankowe etc.) na zestaw pisany tłustym drukiem. Znaki pozostają te same, tylko grubsze. Program (przeznaczony na Spectrum) korzysta z procedury w kodzie maszynowym (29 bajtów). Tak, tak: wczytanie do pamięci tych 29 liczb występujących w segmencie DATA spowoduje, że wszystko co będziemy chcieli napisać używając PRINT, LIST, COPY, LPRINT lub LLIST, będzie pisane tłustym drukiem!

Realizując nasz program, komputer kopiuje – począwszy od adresu 56576 – 96 znaków (768 bajtów) alfabetu języka BASIC, przesuwając je o jedną "kropkę" w prawo. W ten sposób "nowy" zestaw znaków jest przesunięty względem "starego" o jeden pixel. Jednocześnie nakazuje pisanie obu zestawów na raz, co daje efekt tłustego druku.

RYSZARD NIŻNIK

```
  5>REM TŁUSTY DRUK
 10 CLEAR 56575
 20 LET P=57786: LET S=0
 30 RESTORE
 40 FOR I=P TO P+28
 50 READ A: POKE I,A
 60 LET S=S+A
 70 NEXT I
 80 IF S<>2451 THEN PRINT ''
 "ZŁE DANE! PROSZE POPRAWIC."''
': LIST 200: STOP
 90 PRINT "TAK TO WYGLADA PRZED
..."
100 RANDOMIZE USR P
110 PRINT '''A TAK PO WYKONANIU
PRGRAMU."
200 DATA 17,0,221,213,1,0,3,42,
54,92,36,126,167,31,182,18,35,
19,13,32,246,16,244,225,37,34,
54,92,201
```

```
PO WYKONANIU PROGRAMU NA EKRANIE
POWINNO POJAWIC SIE :

TAK TO WYGLADA PRZED...

A TAK PO WYKONANIU PRGRAMU.
```

```
POWROT DO STANDARDOWEGO ZESTAWU:

     POKE 23607,60

(CO SIE STANIE GDY ZAMIAST 60,
WSTAWISZ 56 LUB 88 ?)
```

POLSKIE LITERY W IBM PC

Wykorzystywanie w praktyce programów przetwarzania tekstów nasuwa konieczność używania znaków charakterystycznych dla alfabetu języka, jakim się posługujemy. Trudno sobie wyobrazić pisanie listów, podań, sprawozdań po polsku bez ą, ć, ę, ń, ł, ó, ś, ż, ź. Litery te są elementami naszego alfabetu i nie można ich pominąć.

Używane na naszym rynku komputery pochodzą ze strefy anglojęzycznej i takich liter nie posiadają. Powoduje to komplikacje przy korzystaniu z programów przetwarzających polskie teksty. Można temu zapobiec stosując w komputerach IBM PC lub ich odpowiednikach rozwiązanie proponowane przez pracowników Instytutu Fizyki Polskiej Akademii Nauk.

Rozwiązanie to polega na wprowadzeniu zmiany sprzętowej w karcie podstawowej systemu PC. Zmiana pozwala uzyskać z klawiatury niezbędne w języku polskim ą, ć, ę, ń, ł, ó, ś, ż, ź. Przeróbka karty podstawowej nie powoduje utraty żadnych dotychczasowych jej własności i funkcji. Komputer zyskuje 9 nowych znaków literowych. Wywołanie "nowych" liter z klawiatury odbywa się przez równoczesne naciśnięcie dwóch klawiszy, podobnie jak ma to miejsce w maszynie do pisania. Należy tu podkreślić, że normalna klawiatura komputera IBM PC lub jego odpowiednika nie jest przerabiana. Na 9 klawiszy nanoszone są tylko diagramy ich dodatkowych znaczeń (trzecia funkcja klawisza). Dodatkowe litery umieszczone są po prawej stronie klawiatury na klawiszach zawierających znaki przestankowe i podstawowe znaki graficzne. Elementy sprzętu obsługujące klawiaturę "widzą" ją rozszerzoną o 9 dodatkowych kodów, mających swój obraz graficzny w matrycy znaków wyświetlanych na ekranie monitora.

Kształt "nowych" liter może być dowolnie ustalony przez użytkownika przerobionego systemu PC. Pozwala na to graficzny program definiujący wygląd liter i znaków używanych przez komputer. Opracowane w ten sposób znaki zakodowane są na stałe w matrycy znaków komputera.

Zmiany sprzętowe proponowane przez pracowników Instytutu wprowadzają polskie litery do systemu PC niezależnie od używanego oprogramowania. Większość programów użytkowych dla komputerów PC bez zastrzeżeń współpracuje z rozszerzoną matrycą znaków. Chodzi tu głównie o programy redagowania i przetwarzania tekstów. Przykładowo program WORDSTAR pracuje z przerobionym PC jak z oryginalnym. Zmian wymagają programy sortujące. Oryginalne oprogramowanie tego typu nie poradzi sobie z prawidłową kolejnością pełnego polskiego alfabetu. Nie używane dotąd litery (ich kody są dla programu niezrozumiałe) mogą pojawiać się przypadkowo lub powodować błędy działania programu.

Konsekwencją sprzętowego wprowadzenia polskiego liternictwa do systemu PC jest konieczność dopasowania drukarki współpracującej z systemem. Sposób współpracy, organizacja wydruku, niezbędne zmiany wewnątrz drukarek są również opracowane przez pracowników Instytutu Fizyki.

Obecnie twórcy rozwiązania załatwiają formalności związane z uzyskaniem zastrzeżenia wzoru użytkowego na swoje opracowanie. Należy życzyć sympatycznym konstruktorom pomyślnego załatwienia formalności oraz wszystkim użytkownikom systemów PC, aby ich komputery pisały wszystkie polskie litery.

ZENON RUDAK

CX-TEXT

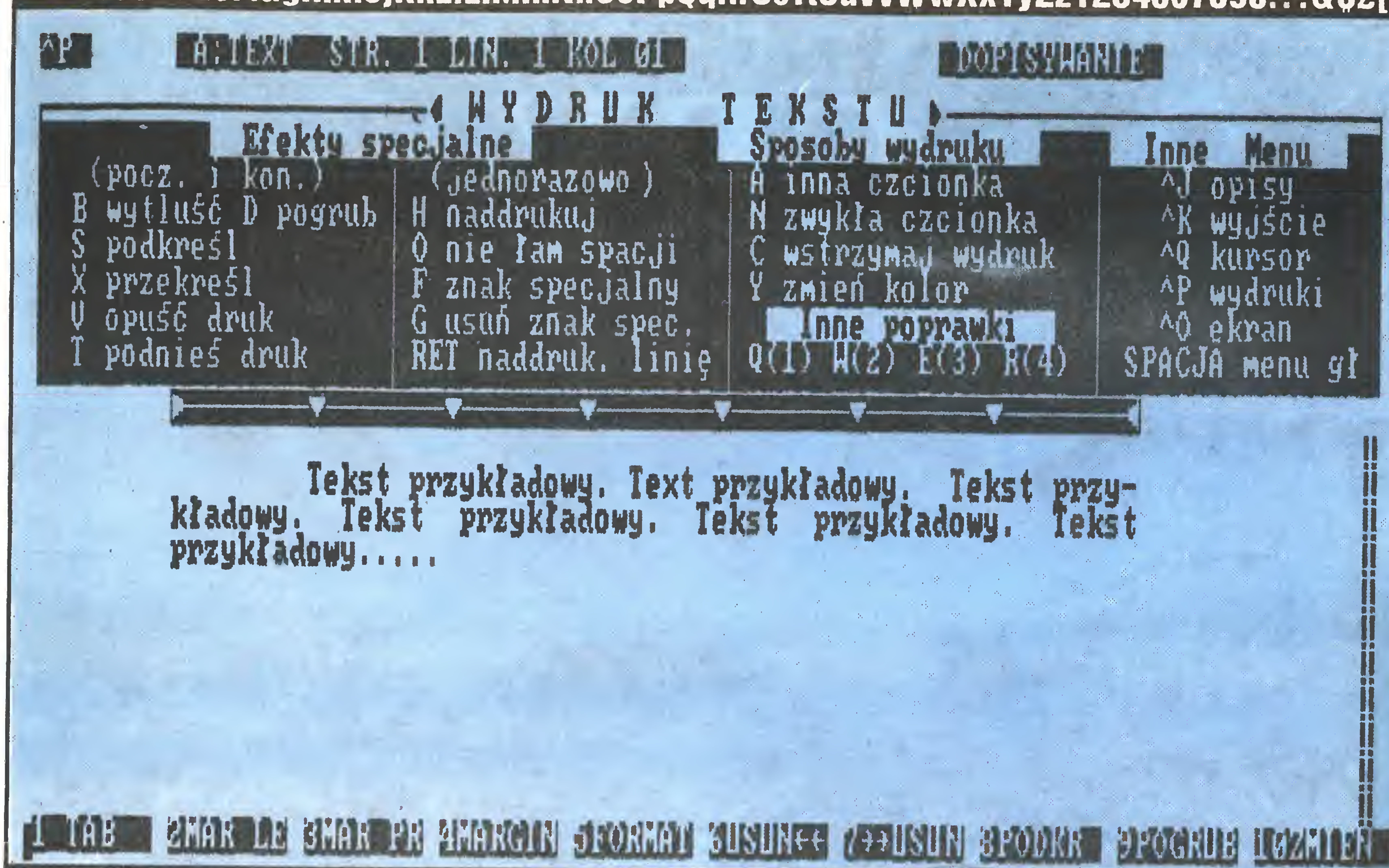
Od pewnego czasu można zauważyć w biurach i przedsiębiorstwach tendencję do zastępowania maszyn do pisania przez zestawy komputer-drukarka. Przyczyną tego jest spora już dostępność mikrokomputerów oraz bogatego oprogramowania służącego do obróbki tekstów. Jednym z najlepszych na świecie programów tego typu jest WordStar. Ma on jednak poważną wadę, która wyklucza możliwość użycia go w typowym polskim biurze – nie umożliwia wprowadzania tekstów zawierających polskie znaki diakrytyczne: ą, ę, ó, ź, ń itp. Problem ten rozwiązano w PPZ CompuTeX opracowując program w pełni odpowiadający funkcjonalnie WordStar'owi, lecz w wersji polskiej. CX-TEXT, bo taka jest jego nazwa, może być zainstalowany na każdym komputerze zgodnym z IBM PC/XT. Klawiaturę komputera program przedefiniowuje tak, aby rozkład klawiszy odpowiadał polskiemu standardowi maszyn do pisania. Jest to istotna zaleta CX-TEXT'u, szczególnie cenna dla maszynistek piszących wiele lat na zwykłych maszynach i zamieniających je na komputer biurowy. Docenią one także drugą bardzo ważną cechę. Program od momentu uruchomienia "pomaga" nam w wykorzystaniu jego dużych możliwości. Przede wszystkim podczas pracy nie musimy mieć pod ręką podręcznika. CX-TEXT przez cały czas wyświetla nam spis możliwych komend, nienaganną polszczyzną zadaje pytania o ewentualne parametry lub informuje o popełnionym błędzie. Dodatkowo w każdej chwili można wywołać funkcję opisującą w sposób wręcz podręcznikowy, co należy zrobić, aby uzyskać dany efekt. Wszystkie teksty, opisy, pytania itp. są przedstawione w języku polskim.

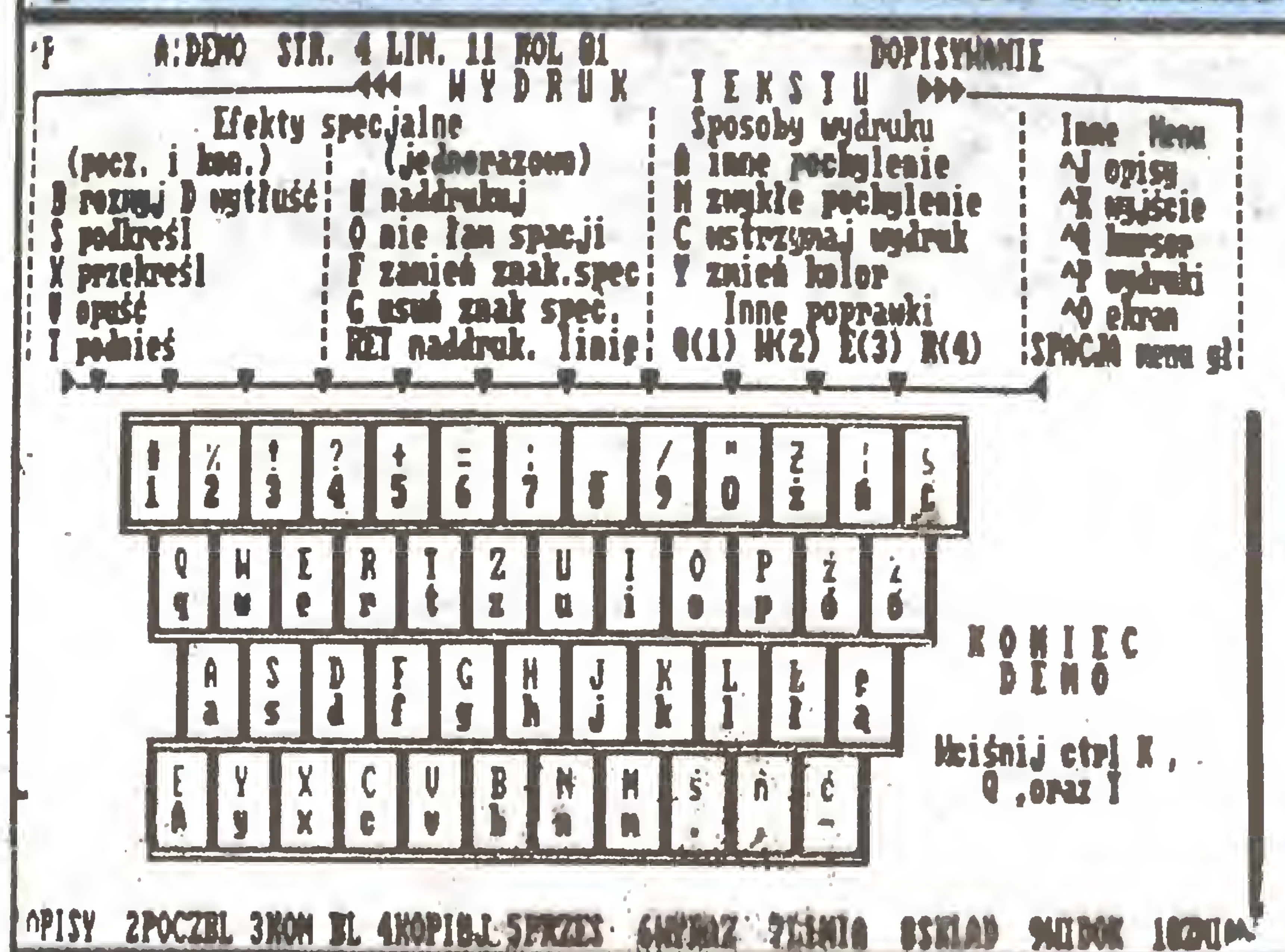
"Polskość" programu jest bardzo ważną, ale nie jedyną zaletą CX-TEXT'u. Ma on zwarty, hierarchiczny układ poleceń, w którym nawet niewprawny użytkownik uczy się szybko poruszać. Struktura zestawów poleceń jest trzypoziomowa. Pierwszy poziom włącza się w momencie wejścia do programu. Umożliwia on wykonanie wielu operacji na zbiorach dyskowych, lecz przede wszystkim pozwala utworzyć dwa typy zbiorów tekstowych. Jeden z nich zwany "standardowym" posiada dodatkowe kody sterujące, dzięki którym program potrafi wykorzystać wszystkie funkcje dotyczące formatowania i wydruku tekstu. Otwierając drugi typ pliku mamy możliwość pisania pod tym edytorem programów, które będą potem na przykład bez przeszkód kompilowane, gdyż CX-TEXT nie umieści w nich swoich znaków sterujących.

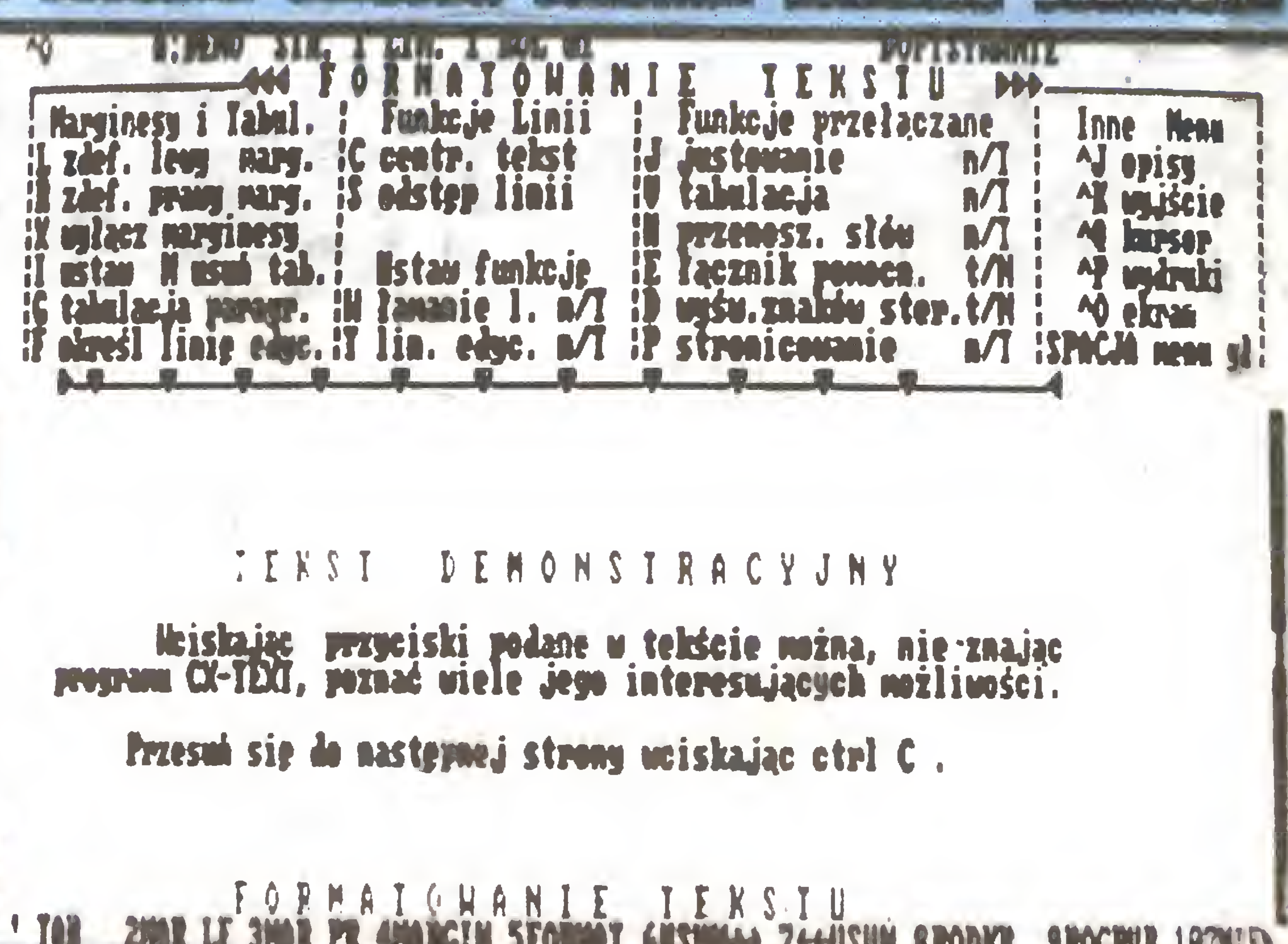
Po otwarciu zbioru znajdujemy się na drugim poziomie poleceń umożliwiającym poruszanie się po tekście i reformatowanie go. Oprócz tego mamy tu możliwość przejścia do jednego z pięciu spisów poziomu trzeciego. Pierwsze z menu zawiera wspomniany wyżej książkowy opis poszczególnych procedur CX-TEXT'u, a następne – polecenia dotyczące odpowiednio: operowania blokami tekstu i zbiorami, sposobów szybkiego poruszania się po tekście, sterowania wydrukiem oraz ustawiania atrybutów kontrolujących wprowadzanie tekstu.

Ważne jest, że dla uzyskania dowolnego polecenia CX-TEXT'u wystarczy wcisnąć jeden lub dwa klawisze. Oprócz typowych dla każdego edytora pełnoekranowego cech takich, jak: możliwość poruszania się kursorem i wpisywanie tekstu na całym ekranie, ustawiania w dowolnych miejscach marginesów i tabulatorów, kopiowania bloków tekstu itp., program ma wiele dodatkowych funkcji.

AaBbCcDdEeFfGgHhIiJjKkLlŁłMmNnOoPpQqRrSsTtUuVvWwXxYyZz1234567890.?!&$£[]





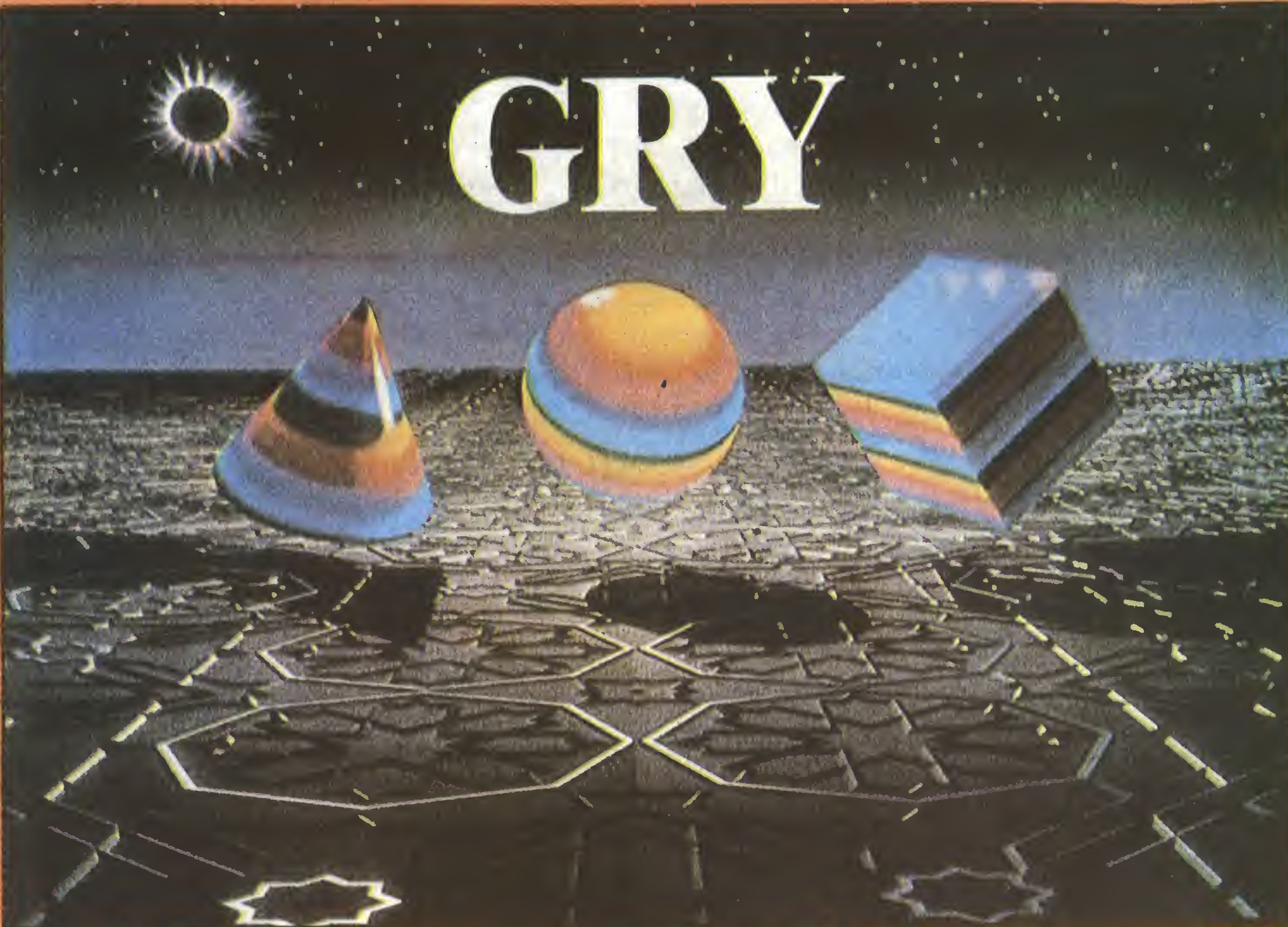


Jedną z nich jest rozbudowane polecenie szukania podanego tekstu i zamiany go na inny. Program może na przykład odnaleźć wszystkie wyrazy, dla których część liter podamy, a część uważamy za nieistotną. Potrafi wtedy wskazać na słowa "lupa", "lipa", "lepa" itp. Jeśli chcemy, przy każdym z nich program może pytać nas o pozwolenie zamiany na inny wyraz. Podobnie rozszerzone są funkcje działające na zaznaczonych partiach tekstu. Blok taki można przy użyciu prostych poleceń przesuwać w inne miejsce, kopiować, usuwać czy nawet zapisać na dysk w postaci osobnego pliku. Wyłania się tu jednak pewna niedogodność programu. CX-TEXT może "przerabiać" tak duży tekst, jaki tylko zmieści się na dysku, trzymając w pamięci operacyjnej wyłącznie bufor zawierający fragment całości. Bufor ten stanowi przemieszczalne "okno" w tekście. Powoduje to spowolnienie opisanych powyżej operacji, gdyż program wiele razy musi czytać interesujący nas plik.

CX-TEXT to jednak nie tylko rozbudowany edytor. Jego "drugi człon" to formater tekstu. W praktyce oznacza to, że wpisując znaki dokumentu nie musimy, w odróżnieniu od maszyny do pisania, dbać na bieżąco o wygląd końcowy wprowadzanego tekstu. Program sam wyrównuje lewy i prawy margines, zmienia stronę we właściwym miejscu, kształtuje odpowiednio akapity oraz nagłówki i stopki stron. Umożliwia nam też uzyskanie praktycznie dowolnego sposobu wydruku i powielania tekstu. Sterowanie drukarką to jednak zupełnie osobne zagadnienie dla CX-TEXT'u. Praktycznie wszystkie typy drukarek mogą we współpracy z tym programem zapewnić takie efekty o jakich użytkownicy zwykłych maszyn do pisania mogą tylko marzyć. Pogrubianie, rozszerzanie druku, zmiana wysokości i kroju liter oraz koloru taśmy w trakcie pracy to tylko część możliwości. Dochodzą do tego tzw. polecenia z kropką, które są wprowadzane jawnie do tekstu, lecz nie są drukowane. Używając ich możemy na przykład łatwo zmieniać marginesy, długości stron wydruku, przesunięcia dla stron nieparzystych (potrzebne przy zszywaniu kartek wydruku) itp.

Na zakończenie warto podkreślić wysoką obecnie cenę czasu pracy. Wyobraźmy sobie sekretarkę wypisującą mozolnie na maszynie zawiadomienia, w których zmianie ulega tylko nazwisko, lub też tony różnego rodzaju formularzy zalegających magazyny i porównajmy to z możliwością włożenia do komputera dyskietki, na której są dziesiątki wzorów zawiadomień, ogłoszeń i formularzy, zmianie kilku zdań i drukowanie tego w potrzebnej ilości egzemplarzy. Nie dziwmy się więc, że ludzie powoli odchodzą od maszyn do pisania i skłaniają się w kierunku coraz bardziej użytecznych komputerów.

ANDRZEJ PŁACZKOWSKI



- WŁADCA PIERŚCIENI
- DUN DARACH
- DRAGONTORC [4]
- POKE: DYNAMITE DAN, SABOTEUR, ATIC-ATAC, HIGHWAY ENCOUNTER...

OCZAMI PILOTA:
- FLIGHT SIMULATION
- FIGHTER PILOT

WŁADCA PIERŚCIENI

Stali czytelnicy tej rubryki (po trzech numerach chyba już mamy stałych czytelników?) zauważyli zapewne, że dość starannie unikamy omawiania programów typowo zręcznościowych. Skłaniamy się raczej ku takim, w których zręczność manualna odgrywa drugo-, a najczęściej trzeciorzędną rolę. Recenzja, którą w tej chwili czytacie, stanowi jasny dowód, że nie zamierzamy z tej drogi zawracać, a nawet wręcz przeciwnie. "Władca Pierścieni" należy do typowych programów przygodowych, opierających się niemal wyłącznie na tekście.

Trylogia J.R.R. Tolkiena "Władca pierścieni" jest utworem, który obrósł już niemal w legendę. Dziełem, które pozycję klasycznego w swym gatunku zdobyło nie tyle dzięki uznaniu krytyki (choć i tego mu nie brakowało), ile dzięki niemal bezprecedensowej fascynacji czytelników, dzięki wpływowi, jaki ma ono na wyobraźnię, zarówno w trakcie lektury, jak i po jej zakończeniu. Nic więc dziwnego, że firma Melbourne House zdecydowała się finansować prace, związane z przeróbką trylogii Tolkiena na grę komputerową. Dodatkowym bodźcem był tu na pewno ogromny sukces programu "Hobbit", dobrze znanego również miłośnikom gier przygodowych w naszym kraju. Nie łudzono się zapewne, i słusznie, że ogromne bogactwo pierwowzoru uda się wcisnąć w 48KB pamięci ZX Spectrum. Jednak o wysiłkach programistów w tym kierunku świadczą rozmiary gry "Władca Pierścieni" – całość, gdy zostanie ukończona, będzie się składała z 3 części, z których każda zawiera 2 programy o pełnej długości. Dotychczas została ukończona i skierowana do sprzedaży tylko pierwsza część. Druga ma się ukazać tuż przed tegorocznymi świętami Bożego Narodzenia, o trzeciej jeszcze nic nie wiadomo.

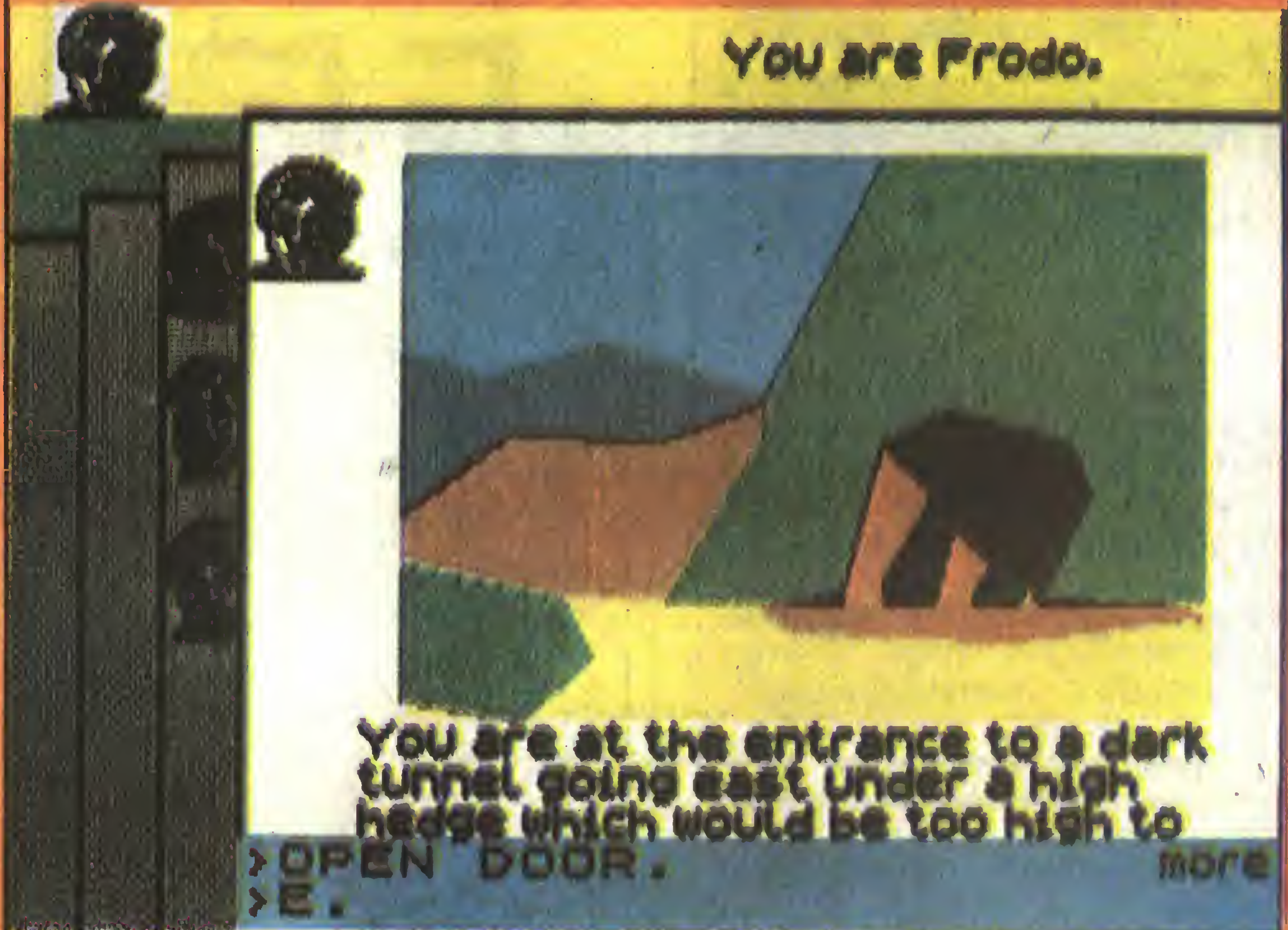
Tak więc na razie musimy zadowolić się jedną trzecią całości. To i tak są dwa programy. Pierwszy z nich rozpoczyna się od wyruszenia Hobbitów z Bag End i powinien zostać zakończony przeprowadzeniem ich przez bród przed Rivendell. Drugi rozpoczyna się w Rivendell, obejmuje wędrówkę przez Morię i Lothorien, aż do rozstania, podczas którego Frodo i Sam udają się do Mordoru, zaś Merry i Pippin na wojnę na południe. Oba programy mogą być rozgrywane niezależnie od siebie, jednak droga z Rivendell na wschód jest znacznie trudniejsza, jeżeli bohaterowie nie posiadają rzeczy, które można znaleźć w pierwszej części.

Program wita nas pytaniem o to, którym z Hobbitów chcemy być – możliwa jest dowolna kombinacja, od jednego do wszystkich. Jednak stopień trudności rozgrywki jest wprost proporcjonalny do ilości postaci, którymi bezpośrednio sterujemy. Wyobrażam sobie (gdyż nie sprawdzałem), że ukończenie choćby pierwszej części, mając na głowie wydawanie szczegółowych poleceń po kolei wszystkim czterem Hobbitom, jest nie lada wyczynem. Niewątpliwie najbardziej zalecane jest sprawowanie bezpośredniego nadzoru nad Powiernikiem Pierścienia, Frodem. Jednak możliwość kontrolowania czterech postaci pozwala na nieczęsty w programach przygodowych wariant rozgrywki – na grę zespołową, w kilka osób, z których każda, porozumiewając się z innymi, prowadzi jednego z bohaterów "Władcy".

Program rozumie ponoć ponad 800 słów, zaś interpreter poleceń umożliwia wydawanie wieloczło-

nowych, złożonych rozkazów (np. OPEN BOX, TAKE AXE, GIVE AXE TO SAM). Niewątpliwie wypisywanie tasiemcowych zdań w języku INGLISH (tak jest on określony przez autorów) ma nieodparty urok, jednak odniosłem wrażenie, że program najlepiej się czuje i ma najmniejsze skłonności do popełniania błędów, gdy otrzymuje krótkie, klasyczne polecenia typu "czasownik plus rzeczownik". Interpreter nie zachwyca prędkością, ale jest to zrozumiałe ze względu na wielkość słownika i oczekiwaną złożoność rozkazów. Dość zabawne (momentami stwarzające wrażenie, że komputerowi coś się w procesorze pomieszało) są odpowiedzi, jakie można uzyskać, gdy polecenie jest niejednoznaczne lub zawiera słowa, które program rozumie inaczej niż ty.

Opisy miejsc, do których bohaterowie docierają, są bardzo szczegółowe i często ekran jest całkowicie wypełniony słowami. Walnie przyczynia się do tego jedyna prawdziwie irytująca cecha programu – opisywanie po kolei wszystkich idących z bohaterem postaci i to za każdym razem, gdy przechodzi on do nowego miejsca lub gdy się rozgląda. Jeżeli grupa jest liczna, na przykład po wyjściu z Rivendell, ten proces zajmuje stanowczo za dużo czasu.

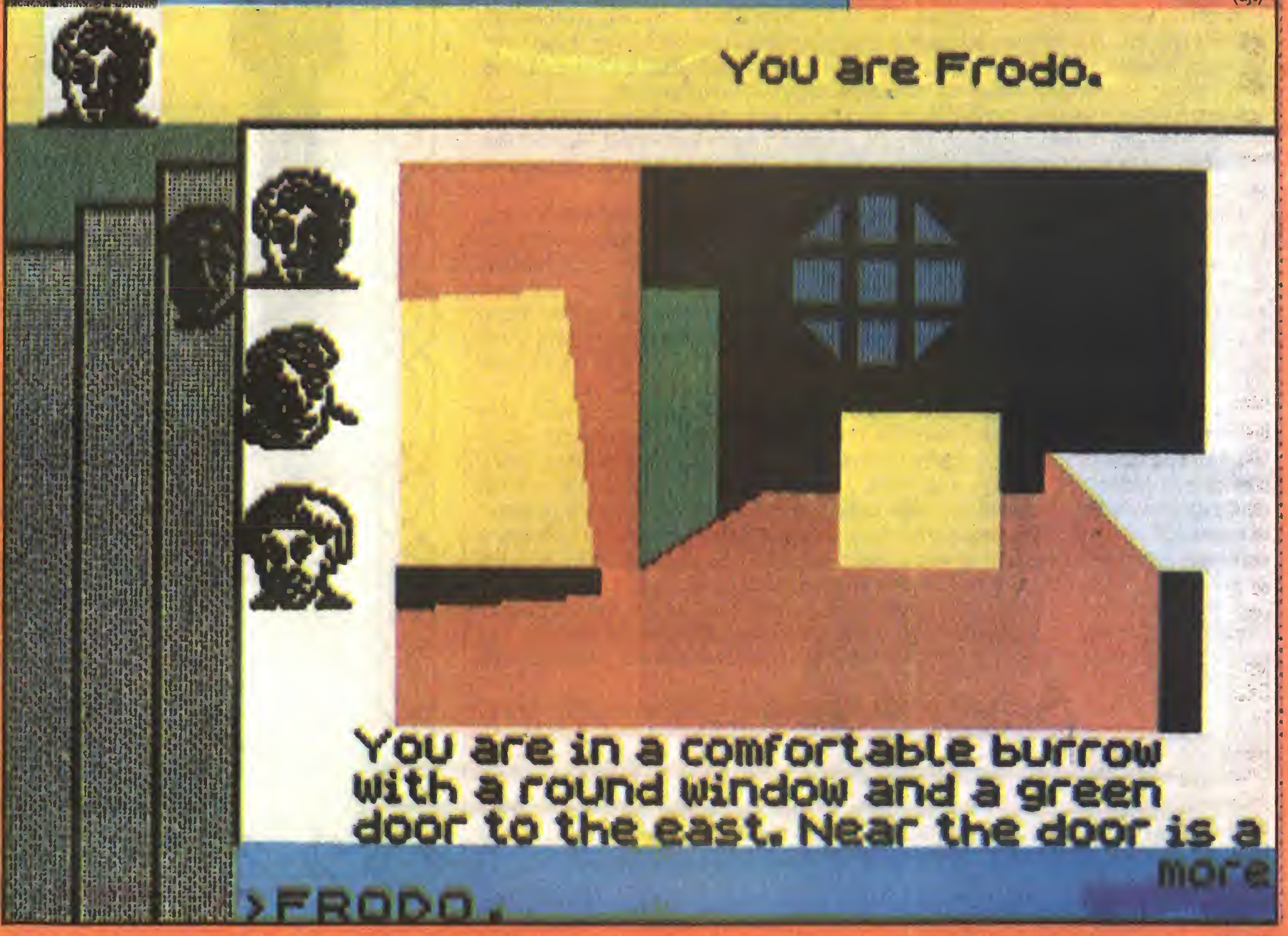


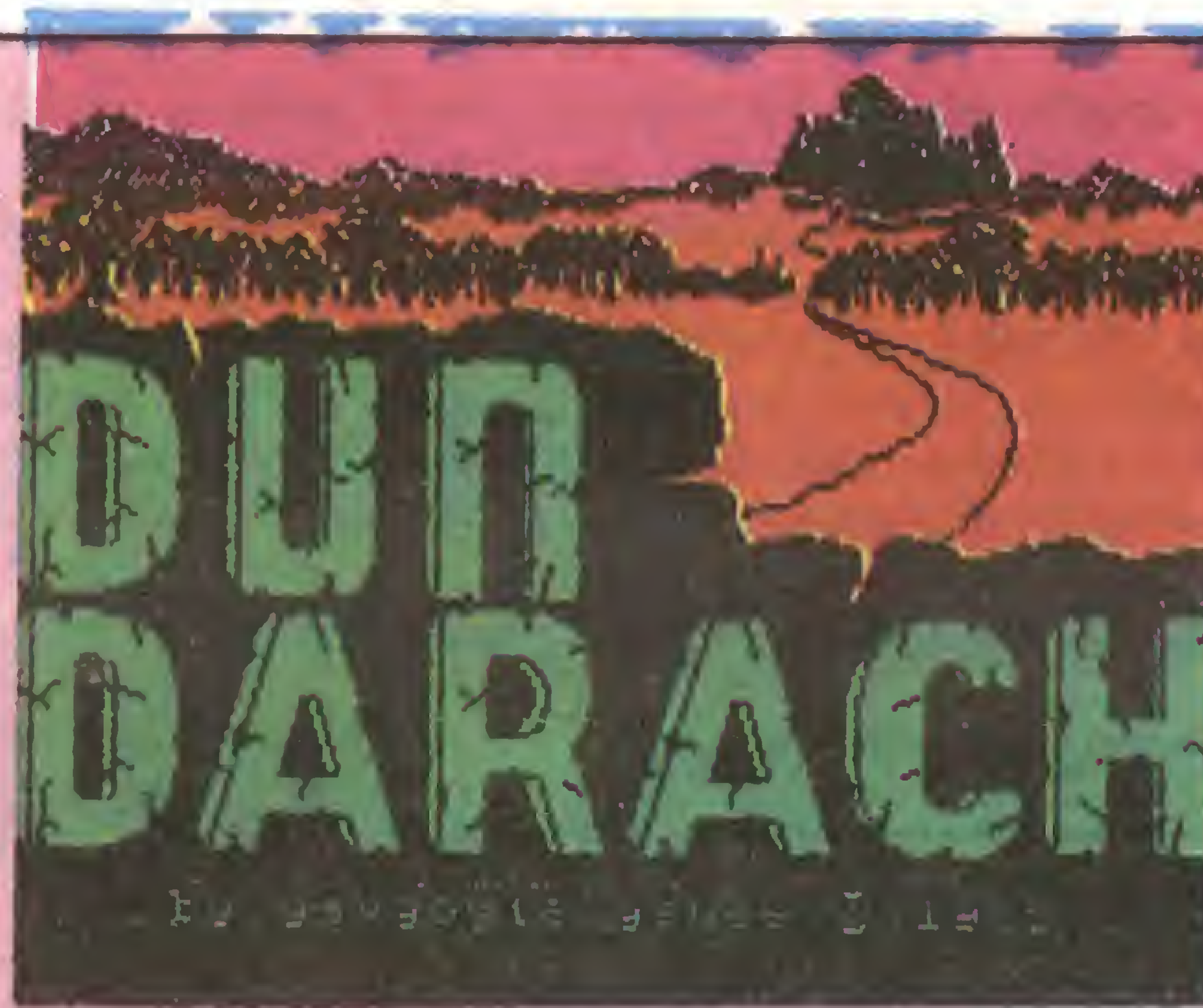
Oczywiście inne postacie, podobnie jak w "Hobbicie" żyją swoim własnym życiem, dzięki czemu jest mało prawdopodobne, aby gra potoczyła się choćby dwa razy w ten sam sposób. Bardzo ważnym elementem jest komunikowanie się z innymi mieszkańcami śródziemia. W niektórych sytuacjach jest to jedyny sposób osiągnięcia zamierzonego celu. Zwrócenie się do osoby, znajdującej się razem z bohaterem, musi mieć formę SAY TO... "..." (np. SAY TO SAM "TAKE SHORT SWORD, GO EAST AND KILL BLACK RIDER").

I jeszcze jedna bardzo ważna rzecz – jednym z ulubionych zajęć oryginalnych Hobbitów Tolkiena było jedzenie. Nie inaczej jest z Hobbitami komputerowymi. Należy pamiętać o tym, aby zawsze posiadali przy sobie coś, czym mogliby się posilić. Konsekwencje wygłodzenia mogą być naprawdę godne pożałowania.

Kilka wniosków końcowych. Nigdy żaden program komputerowy nie zastąpi książki. Choćby był, tak jak "Władca Pierścieni", bardzo długi i niezwykle skomplikowany, choćby starał się wiernie odtworzyć fabułę, charaktery i nastrój. Droga powinna wieść od książki do programu (z ewentualnym pominięciem tego drugiego etapu), nigdy odwrotnie. Gdy o tym będziemy pamiętali, możemy z czystym sumieniem powiedzieć: "To dobrze, że powstają takie gry, jak ta". I w wolnej chwili zasiąść do radosnego stukania w klawisze. Bo zabawa jest przednia.

(djt)





Po krwawej i niepotrzebnej bitwie przeciwko Conachcie, w której zginął jej władca, książę Amhair, dwaj przyjaciele, Cuchulainn i Loeg, wracali wreszcie do domu. Pod koniec drugiego dnia podróży dotarli do przydrożnej gospody, starej i niezbyt dobrze utrzymanej, jednak, sądząc po ilości zaprzęgów na podjeździe, często i licznie odwiedzanej. Wchodząc do środka wymienili kilka uwag na temat stojącego tuż przy drodze dziwnego powozu, zaprzęgniętego w cztery jak noc czarne konie.

Zamówili posiłek i usiedli, czekając aż karczmarz go przyniesie. Chwilę później do ich stołu podeszła uderzająco piękna dziewczyna. Powiedziała, że nazywa się Skar i jest właścicielką stojącego przed karczmą powozu. Prosi o pomoc w naprawieniu pękniętej na którymś z wybojów osi. Loeg zgodził się skwapliwie i w ślad za Skar wyszedł z gospody. Po chwili Cuchulainn również postanowił się przyłączyć, gdy jednak znalazł się przed karczmą, ze zdumieniem stwierdził, że powóz i piękna dziewczyna zniknęły. Zniknął również jego przyjaciel. Jedynie na horyzoncie zanikał obłok pyłu, a wiatr przyniósł dalekie echo upiornego śmiechu.

Cuchulainn domyślił się, że Skar jest czarownicą, stronniczką księcia Amhaira i że porwała Loega do Dun Darach, sekretnego miasta, żeby dokonać na nim zemsty za klęskę Conachty. Jeśli tak, to jego obowiązkiem jest przeszkodzić Skar w ostatecznej realizacji jej planu.

W taki oto sposób nasz znajomy z programu TIR NA NOG, celtycki heros Cuchulainn, został wplątany w kolejną przygodę. Jego zadaniem jest zbadanie miasta Dun Darach i uwolnienie swego druha, Loega. Uda mu się to zadanie spełnić, jeśli rozwiąże piętrzące się przed nim zagadki, znajdzie odpowiednie przedmioty i zaniesie je we właściwe miejsca czy też podaruje osobom, które ich potrzebują.

Miasto Dun Darach podzielone jest na kilka dzielnic, z których każda posiada odmienny charakter i funkcje – np. w dzielnicy rozrywki znajdują się domy gry, a w sektorze finansowym – banki. Wszystkie ulice mają nazwy, zaś wejścia do budynków oznaczone są numerami – ułatwia to w znaczny sposób poruszanie się i poznawanie miasta.

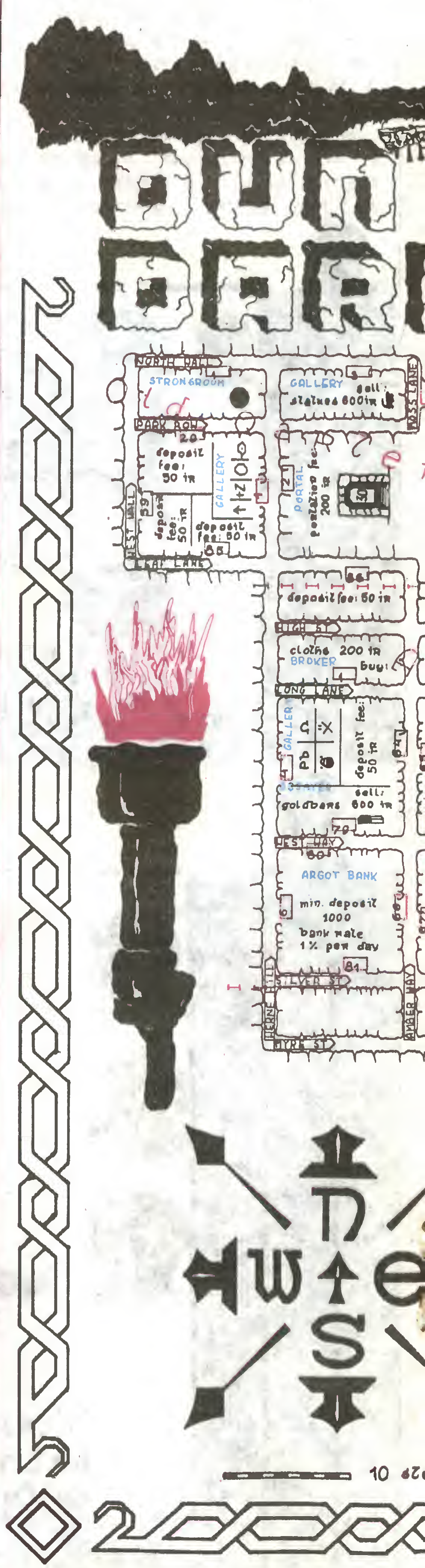
Niezbędnym do osiągnięcia celu elementem postępowania są kontakty ze stałymi mieszkańcami miasta. Mogą one być dwojakiego rodzaju – handel z właścicielami sklepów i kantorów bądź uzyskiwanie informacji od głównych bohaterów przygody w zamian za różnego typu przedmioty. O ile sposób postępowania przy kupnie czy sprzedaży jest dość oczywisty, o tyle porozumienie się z postaciami wędrującymi po ulicach może nastręczać niejakie trudności. Nie tylko trzeba odgadnąć co i komu jest potrzebne, ale również rzecz tę znaleźć. Na dodatek niektórzy mieszkańcy Dun Darach najwyraźniej są złodziejaszkami i potrafią w dość bezceremonialny sposób pozbawić ciężko zarobionych pieniędzy czy znalezionych po wielu trudach przedmiotów. Na szczęście w Bractwie Złodziei można sobie wykupić immunitet od ich lepkich palców.

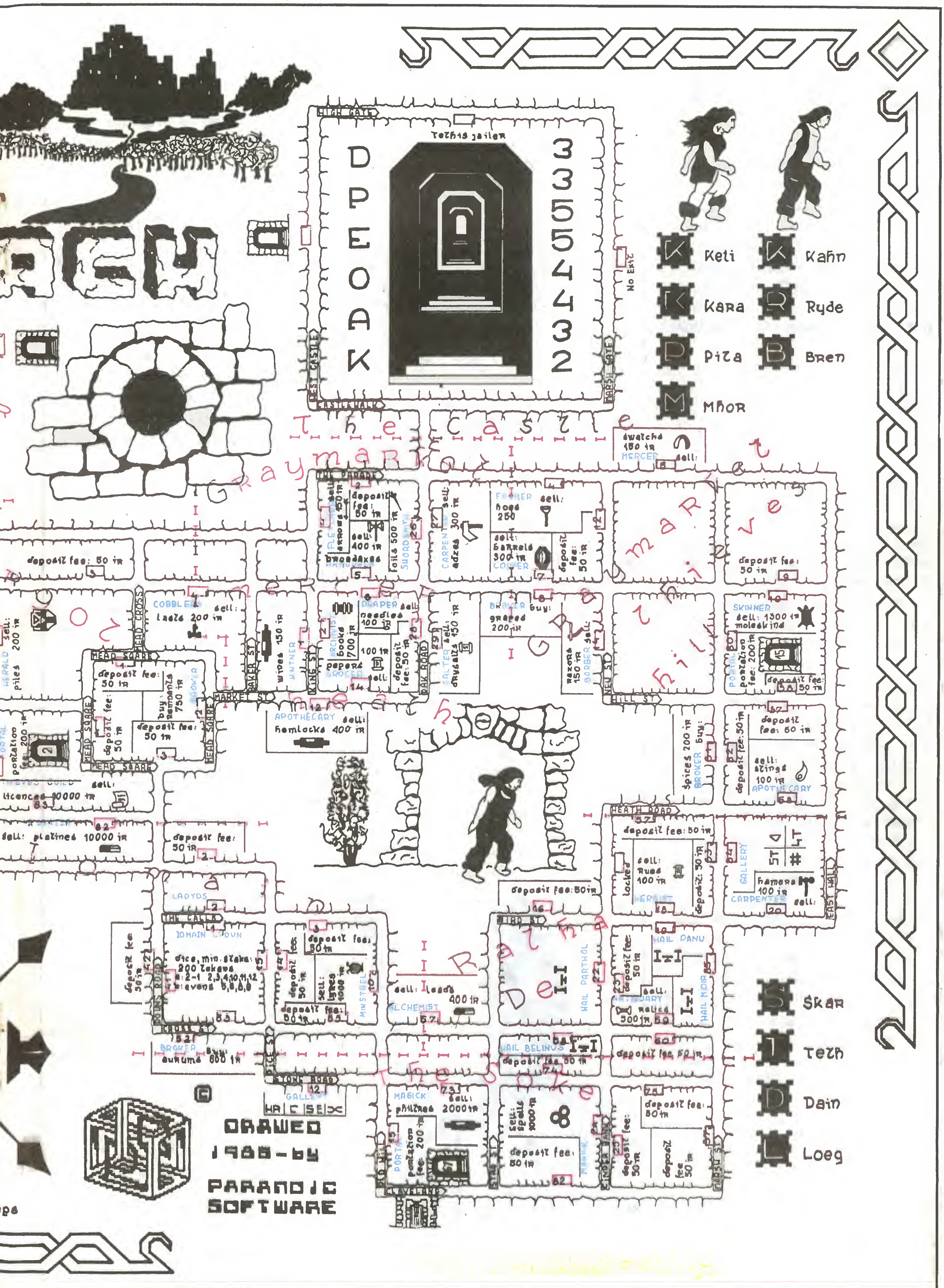
W Dun Darach są nie tylko sklepy i kantory – natkniemy się również na banki, przechowalnie, domy gry, portale-przejścia, magiczne komnaty, świątynie i galerie. Funkcje banków są dość oczywiste – można w nich lokować pieniądze w zamian za odsetki. Przechowalnie służą jako magazyny posiadanych, a chwilowo zbędnych przedmiotów. Domy gry to najszybszy, najłatwiejszy i najprzyjemniejszy sposób zarobienia (oj, nie zawsze!) pieniędzy. Dzięki portalom można szybko przenosić się z jednej części miasta do drugiej. W magicznych komnatach można zakupić pergaminy, dostarczające pożytecznych informacji. Również w świątyniach (oznaczonych HAIL plus imię) można uzyskać podpowiedzi, jednak pod warunkiem, że się tam zaniesie tarczę z imieniem boga, któremu świątynia służy. Najtrudniej jest dowiedzieć się czegoś w galeriach. Na ich ścianach wiszą obrazy, mające jakiś związek (czasem bardzo pokrętny, trzeba przyznać) z przedmiotami, jakie należy tam przynieść. Jeżeli na stole pod obrazem zostanie złożona właściwa rzecz, rysunek w ramach ulega zmianie. W tym przypadku najbardziej skutecznym sposobem postępowania jest niestety metoda prób i błędów.

Nasz dzielny Cuchulainn ma przed sobą naprawdę niełatwe zadanie. Tym większa satysfakcja dla tych, którzy mu skutecznie pomogą. Satysfakcja na końcu, a przedtem – dzięki pomysłowości i przewrotnemu poczuciu humoru autorów programu – kilka (-naście, -dziesiąt) godzin dobrej komputerowej zabawy.

Klawisze sterujące to: ruch w lewo i w prawo – od "Z" do "SYMBOL SHIFT" naprzemiennie; zmiana "ustawienia kamery" – od "A" do "L"; wejście w drzwi – "ENTER"; podniesienie i wyrzucenie niesionego przedmiotu – od "Q" do "P" naprzemiennie; wybór jednego z niesionych przedmiotów – "2", "3", "7", "8" i "9"; i wreszcie zaoferowanie przedmiotu – cztery klawisze narożne.

djt)





DPEOAK
3355432
Tezhis jailer
HIGH GATE
NO EXIT
Keli
Kahn
Kara
Ryde
Piza
Bren
Mhor
The Castle
Graymarket
market
THE PARADE
FARMER sell: hose 250
CARPENTER sell: adzes 300 IR
COOPER sell: barrels 300 IR
deposit fee: 50 IR
COBBLER sell: laces 200 IR
DRAPER sell: needles 100 IR
SKINNER sell: 1300 IR moleskind
APOTHECARY sell: hemlocks 400 IR
MARKET ST
HEAD SQARE
HEATH ROAD
LADYDS
DOMAIN LADUN
dice, min. stake: 200 tokens
ALCHEMIST
HAIL PARTHOL
HAIL DANU
HAIL BELINUS
MAGICK sell: philtres 2000 IR
STONE ROAD
EAST WALL
Skar
Tezh
Dain
Loeg
DRAWED 1986 - by
PARANOIC SOFTWARE

Dragontorc [4]

Witajcie w TROLLSTONES. Jestem pewien, że dotarcie tutaj z miejsca, w którym pożegnaliśmy się miesiąc temu, nie sprawiło nikomu kłopotu. Tym razem mam zamiar wreszcie doprowadzić Was do końca, do uwolnienia Merlina. A więc do dzieła.

Po uporaniu się z leśnymi diabełkami (pomoże Wam w tym wojownik – WARRIOR) przeszukajcie liście – znajdziecie czar BANE. Z pnia wyjmijcie złotą monetę i wrzućcie ją do stawu. Otworzycie w ten sposób drogę do Przedsionka Piekieł (HELLSMOUTH), wiodącą przez dno stawu.

Przedsionek Piekieł zamieszkiwany jest przez nietoperze, których można się pozbyć za pomocą MISSILE oraz przez węże, przed którymi najlepszą obronę stanowi pochodnia. Znajdźcie siekierę i użyjcie jej do rozbicia jaj węży. W jednym z nich znajdziecie klucz do skrzyni. Otwórzcie ją i wyjmijcie symbol korony. Teraz musicie wykazać się dużą zręcznością – umieśćcie ten symbol na obłożonym klątwą kluczu, unikając kontaktu z ognistymi kulami. Weźcie ten klucz i otwórzcie nim zamknięte drzwi. Gdy pojawi się demon, zniszczcie go, używając czaru DEMON. Następna korona jest Wasza.

Zabijcie latające oczy czarem BANE i odsłońcie wyjście, kładąc symbol korony na kamiennej płycie. Udajcie się do Świątyni Halgora i zniszczcie koronę.

Ostatnią z czterech koron można znaleźć w Przeklętej Krypcie (CURSED CRYPT). Żeby tam się dostać, musicie dotrzeć do TORHENGE. Po lesie (GREATWOOD) włóczy się patrol goblinów, który znalazł tam skrzynię, pilnowaną pzez skrzydlatego demona. Przed jego czarami ochroni Was MANTLE. Złapcie królika i włóżcie go do jednej z dwóch znajdujących się obok siebie dziur. Znalezionym kluczem otwórzcie skrzynię. Zabierzcie z niej symbol Ankh i ponownie złapcie królika. Tym razem pozwólcie mu wejść do pojedynczej dziury. Ucieknie z niej wąż, a Wy będziecie mogli zabrać znajdujący się tam czar. Teraz musicie nakarmić królika marchewką. Jedząc, wygrzebie on z ziemi kamienną płytę. Dotknijcie jej symbolem Ankh, a odsłoni się wejście. Skorzystajcie z niego, zabierając obłożony klątwą klucz (CURSED KEY), symbol Ankh i pochodnię.

Z upiorami, zamieszkującymi Przeklętą Kryptę, poradzicie sobie dzięki czarowi BANE. Niebezpieczeństwo, czyhające w pokoju na lewo od tego, przez który tu weszliście, można wyeliminować dotykając symbolu Ankh na ścianie tym, który Wy posiadacie. Wiszący symbol zmieni kształt i upadnie na podłogę. Weźcie go. Podobnie postąpcie w dwóch z kilku następnych komnat, za każdym razem dopasowując odpowiednie symbole. W ten sposób odkryjecie wyjście na zewnątrz.

Obłożony klątwą klucz (CURSED KEY) otwiera większość ze znajdujących się w Przeklętej Krypcie zamkniętych drzwi. Uwaga na niewidzialne węże – ochroni przed nimi czar MANTLE, zaś ze skutków ich ukąszeń leczy HEAL. Weźcie mały, niebieski kluczyk i otwórzcie nim niebieską skrzynię. Znalezionym w niej kluczem otwórzcie ostatnią parę jeszcze zamkniętych drzwi. Przeszukajcie dwa katafalki, znajdujące się w dalszych komnatach. W jednym z nich jest czar, a w drugim korona.

Wróćcie do Świątyni Halgora i z tą koroną postąpcie tak samo, jak ze wszystkimi poprzednimi. Teraz dotknijcie Kryształem Zamroczenia (CRYSTAL OF ANTITHOUGHT), tego, co kiedyś było kowadłem. Kryształ się zmieni i otrzymacie wiadomość. Wróćcie do Przeklętej Krypty, zabierając Kryształ ze sobą. Dotknijcie nim wędrującej dłoni, a gdy znieruchomieje, zabierzcie ją.

Teraz, po zniszczeniu czterech koron, jesteście już gotowi do ostatecznej rozprawy z czarownicą Morag. Udajcie się na północ, do WYRMWOOD. Jesteście już tak zaprawieni w bojach, że zamieszkujące tę puszczę potwory nie powinny stanowić dla Was problemu. Przeszukajcie wszystkie stawy, liście itd. Znalezione brylanty podarujcie elfom, a otrzymacie w zamian pochodnię, czar MISSILE i klucz. Tym kluczem otwórzcie skrzynię, stojącą w drugim z łatwo dostępnych pomieszczeń Zamku Morag (CITADEL OF MORAG). Znajdziecie w niej czar DETECT. Zapalcie pochodnię od ogniska goblinów oraz otwórzcie stojącą obok niego skrzynię. Znajduje się w niej łuk dla ostatniego z elfów oraz klucz, otwierający główne wejście do Zamku. Wypracujcie sobie sposób omijania unoszącego się w powietrzu sześcianu. Zabijcie przynajmniej jednego ze strzegących wejścia goblinów, gdyż w przeciwnym razie elfy nie będą chciały pójść za Wami do Zamku. Gdy już tam wejdą, zajmą się pozostałymi goblinami. To jest generalna zasada – tam, gdzie to możliwe, pozwólcie elfom walczyć za Was. Otwórzcie następne drzwi tym samym kluczem, wejdźcie i poczekajcie na elfy (strzeżcie się ognistych kul). Użyjcie klucza po raz trzeci, zabijcie nietoperza i otwórzcie drzwi po lewej stronie (znów ten sam klucz). W dwóch pokojach na prawo niewidzialne gobliny strzegą skrzyń – rozproszcie ich magię czarem DETECT. Weźcie klucz i otwórzcie obie skrzynie. Wyjmijcie z nich czar UNDO i pentagram – powinien on zostać umieszczony na odpowiadającym mu symbolu w jednej z dalszych komnat. W ten sposób otworzy się kolejne wyjście na zewnątrz, poza Zamek. Zostawcie pentagram i klucz w tej komnacie. Strzeżcie się skrzydlatego demona, odbierającego zdolność otwierania drzwi – wykorzystajcie czary MANTLE i MISSILE, żeby go zniszczyć.



Droga do czarownicy Morag wiedzie przez pokój, w którym zamieszkuje diabełek (nie dotykajcie go, gdyż zostaniecie przeniesieni poza Zamek). Spotkane po przejściu tego pokoju demony i szkielety starajcie się niszczyć sami, gdyż wydają się one być zbyt silne dla elfów. Znajdźcie staw, w którym będziecie mogli uzupełniać swą energię. Zielony kluczyk, na który natraficie w jednym z pomieszczeń, jest zaczarowany – wychodzą z niego węże. Połóżcie go na podłodze, a potem użyjcie pochodni do ochrony przed wężami i czaru UNDO do zdjęcia z niego uroku. Gdy już będzie się nadawał do użytku, otwórzcie nim skrzynię. Zabierzcie z niej biały klucz, zaś pentagram zostawcie na później. Otwórzcie zamknięte drzwi (za nimi znajduje się czarownica Morag) i pozwólcie goblinom gonić Was tak długo, aż spotkają się one z elfami. W komnacie, w której przebywa Morag, użyjcie czarów SLOW lub UNSEEN. Połóżcie dłoń zabraną z Przeklętej Krypty na wiszącej ponad głową czarownicy koronie. Korona spadnie. Zabierzcie ją oraz różdżkę i idźcie na lewo. Użyjcie BANE, żeby pozbyć się upiorów i dotknijcie różdżką wyrzutni ognistych kul. Zmieni się ona w klucz, otwierający drzwi, znajdujące się na prawo od komnaty Morag. Zabijcie węże zapaloną pochodnią i przeszukajcie pokój za pomocą różdżki. Znajdziecie skrzynię, otwieraną zielonym kluczykiem. Dotknijcie ją różdżką, żeby rozproszyć ochronną magię i zabierzcie czar MINDKEY. Zanieście różdżkę, koronę i pentagram z powrotem do pokoju, w którym znajduje się sześcian-wyjście i drugi pentagram. Zostawcie tu ten, który teraz przynieśliście i wyjdźcie do sąsiedniej komnaty. Otwórzcie zamknięte drzwi, używając czaru MINDKEY. Otwórzcie skrzynię żółtym kluczykiem (znalezionym w pokoju, strzeżonym przez niewidzialne gobliny), jednak uwaga – ta skrzynia również jest chroniona magią. Użyjcie różdżki, żeby tę ochronę zlikwidować. W skrzyni znajduje się trzeci pentagram. Zanieście go do sąsiedniego pomieszczenia. Połóżcie jeden z pentagramów na sześcianie, a potem jak najprędzej następny. Sześcian stanie się teraz wejściem do CAIRNDOOM, miejsca będącego już ostatnim etapem Waszych poszukiwań. Wchodząc tam zabierzcie ze sobą koronę.

Otwórzcie wszystkie drzwi czarem MINDKEY i znanymi już sposobami rozprawcie się z goblinem, krasnoludem i skrzydlatym demonem. Wejdźcie do pokoju, w którym leży ogromny głaz. Czar MINDKEY rozbije go, uwalniając tego, który Was w to wszystko wplątał – Merlina. Jeszcze tylko podarunek dla niego – korona... i to już koniec.

SUFLER

Długo oczekiwany "DYNAMITE DAN" firmy Mirrorsoft pojawił się wreszcie w naszych zbiorach. Jest to typowa gra platformowa, na pierwszy rzut oka przypominająca "JSW". Przewyższa ona jednak wzorzec lepszą grafiką i większym stopniem trudności. Celem jest uwolnienie przyjaciółki Dana i ucieczka sterowcem. Dan musi przeszukać cylindryczny dom o 48 pomieszczeniach, zbierając po drodze różne przedmioty, z których najbardziej przydatne są laski dynamitu, służące jako broń i pożywienie. Poruszanie się ułatwiają windy, trampoliny, teleportacje i drabinki sznurowe. Jednocześnie na Dana czyhają liczne pułapki i niebezpieczeństwa – właśnie ich usunięciu ma służyć wprowadzenie poniższych zmian. Ich wybór pozostawiamy Tobie, Czytelniku. Jak zwykle będziemy korzystali z pomocy programu "COPY COPY". "Nieśmiertelność" uzyskujemy wpisując POKE 52678,0 oraz POKE 57035,0. Ponadto, pisząc POKE 58770,201 możemy usunąć wszystkie



zagrażające życiu Dana potworki, zaś POKE 59093,201 unieszkodliwia "promienie śmierci". Te poprawki działają tylko w wersji składającej się z trzech segmentów – zetknąłem się z wersją czterosegmentową, której ostatnia, czwarta część wgrywa się w obszar pamięci, przeznaczony do obsługi ekranu, a następnie dokonuje deszyfracji pozostałej części programu. Ta wersja wydaje się być mało elegancka, a ponadto poważnie utrudnia wprowadzanie poprawek.

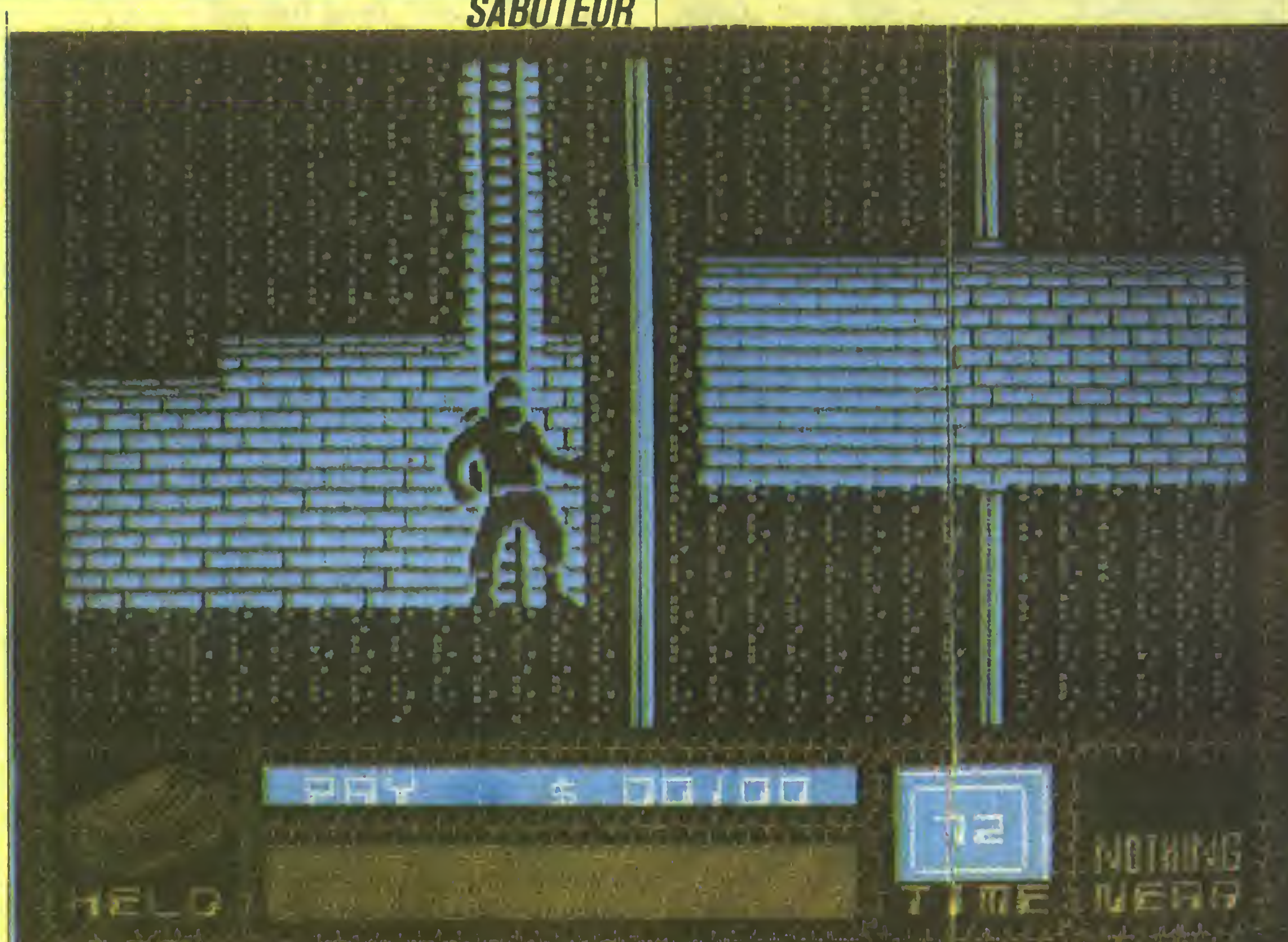
"SABOTEUR" firmy Durrell jest zręcznościową symulacją akcji najemnego komandosa, doskonale wyszkolonego w walce i w prowadzeniu akcji sabotażowych. Jego zadaniem jest odnalezienie dyskietki komputerowej (a jakże), zawierającej listę przywódców rebelii oraz odszukanie bomby i zniszczenie z jej pomocą wrogiej warowni. Po uruchomieniu czasowego zapalnika musi on dostać się na dach, porwać helikopter i wrócić po należną zapłatę. Każda skuteczna akcja wynagradzana jest przez komputer nie jakimiś abstrakcyjnymi punktami, ale twardą walutą. Wyjątek stanowi zabicie psa, programista jest widocznie miłośnikiem zwierząt. W lewej dolnej części ekranu pokazywany jest niesiony przez sabotażystę przedmiot, zaś w prawej pojawiają się przedmioty, znajdujące się w pobliżu. Środkowa dolna część informuje o stanie energii najemnika. Utracone siły regeneruje odpoczynek. Powodzenie akcji zależy od szybkości działania, o czym przypomina tykający zegar. POKE 46998,0 zatrzymuje upływ czasu, zaś POKE 47009,0, POKE 47010,0 i POKE 47011,0 eliminują sprawdzanie, czy przekroczony został jego limit, powodując jednak zakłócenie przy wyświetlaniu cyfr. Najbardziej efektywne okazuje się POKE 29894,0, dający natychmiastową regenerację sił.

Wiele starszych gier do dziś cieszy się dużym powodzeniem, dlatego w skrócie przedstawię poprawki dające "nieśmiertelność" w kilku z nich. A więc: "PSSST" : POKE 24984,0; "ATIC ATAC" : POKE 36519,0; "CAULDRON" : POKE 40060,0; "BOOTY" : POKE 58294,0; „AIRWOLF" : POKE 45982,0; "WRIGGLER" : POKE 50173,0. Zapowiedziane w poprzednim numerze poprawki do "HIGHWAY ENCOUNTER" – POKE 40905,0, POKE 40772,195, POKE 40773,123 i POKE 40774,157 dają całkowitą odporność na kolizje.

Na zakończenie lista najlepszych programów roku 1985 według czytelników angielskiego miesięcznika CRASH. W klasyfikacji generalnej pierwsze miejsce zajął program "ELITE", produkt firmy Firebird, wyprzedzając "WAY OF THE EXPLODING FIST" (Melbourne House) i "STARQUAKE" (Bubble Bus). Wśród zdobywających coraz większą popularność gier zręcznościowo-przygodowych, tzw. arcade adventure, w bardzo silnej konkurencji zwyciężył "FAIRLIGHT" (The Edge), przed "STARQUAKE", "MARSPORT" z Gargoyle Games i "DRAGONTORC" (Hewson Consultants). Ponadto "FAIRLIGHT" uzyskał najwyższe oceny za grafikę oraz walory muzyczne i plastyczne, "STARQUAKE" zaś zaprezentował, zdaniem angielskich czytelników, najlepszą ilustrację dźwiękową. Gry przygodowe zostały podzielone na dwie kategorie. "RED MOON" firmy Level 9 uznany został za najlepszy program przygodowy z tych, które zawierały również ilustracje graficzne. Niższe oceny uzyskały kolejno "DOOMDARK'S REVENGE" i "GREMLINS". Z wyłącznie tekstowych gier przygodowych pierwsze miejsce zajął program "MORDON'S QUEST", a następne "HAMPSTEAD" i "BORED OF THE RINGS". W kategorii gier platformowych zwyciężył "DYNAMITE DAN", za nim ulokowały się "MONTY ON THE RUN" i "NODES OF YESOD". Zwolennicy "strzelaniny" (ang. schoot'em up) wybrali kolejno "COMMANDO", "MOON CRESTA" i "ELITE". Ponadto "TOMAHAWK" uznany został za najlepszą symulację lotu, "HYPERSPORTS" – najlepszą grę sportową, zaś "ARNHEM" – wojenną. Wybrano też największe "niewypały". Tu zwyciężył "THE GREAT SPACE RACE" firmy Legend, wyprzedzając "FLIGHT PATH 737" i "SUPERGRAN".

GRZEGORZ CZAPKIEWICZ

SABOTEUR



SYMULACJE LOTU – OCZAMI PILOTA

W szkoleniu personelu latającego od dawna stosuje się symulatory lotu. Są to kabiny pilota stanowiące kopię prawdziwych kabin. Można dzięki nim imitować niebezpieczne sytuacje; tworzenie ich podczas rzeczywistych lotów byłoby zbyt ryzykowne.

Równolegle z rozwojem komputerów osobistych programy symulacyjne wyszły poza ośrodki szkolenia i zyskały sporą popularność.

Poniżej omówimy najlepsze – jak się wydaje – z nich: powstały dość dawno "FLIGHT SIMULATION" i nieco późniejszy "FIGHTER PILOT".

FLIGHT SIMULATION

Prowadzimy niewielki, jednosilnikowy samolot turystyczny, przeznaczony przede wszystkim do lotów spacerowych. Można nim jednak wykonać kilka prostych figur akrobacyjnych. Obraz na ekranie ukazuje nam przyrządy kontrolne samolotu (rys. 1) i krajobraz widoczny przez szybę kabiny pilota; możemy go także przełączyć "na mapę", z której odczytamy pozycję i kierunek lotu samolotu.

Zanim stery samolotu zostaną oddane w nasze ręce musimy zdecydować, w którym momencie chcemy rozpocząć lot:

1. Gdzieś w powietrzu (in flight)
2. W ostatniej fazie lądowania (final approach)
3. Z pasa startowego z wzniesieniem się w powietrze (take off)

Następnie decydujemy się czy chcemy, by wiał wiatr (Y) czy też nie (N). Ukazuje się tablica rozdzielcza, możemy rozpocząć lot.

Na mapie widać dwa lotniska. Jedno duże, międzynarodowe (MAIN), którego pas startowy usytuowany jest dokładnie w kierunku wschód–zachód (długość pasa wynosi ponad 2 tys. metrów) oraz drugie, mniejsze, przewidziane do przyjmowania samolotów sportowych. Długość pasa startowego usytuowanego w kierunku północ–południe wynosi 700 m. Uwaga: mapa nie oddaje długości pasów startowych.

Na przedłużeniu pasów startowych, w odległości 1 mili od obu ich końców znajdują się ułatwiające lądowanie radiostacje. Dalsze trzy rozmieszczone są nad brzegami trzech jezior. Na mapie migają zawsze ta radiostacja, którą wybieramy naciskając klawisz "R". Przyrządy nawigacyjne będą pokazywać pozycję i odległość liczone od tej właśnie radiostacji.

W prawym górnym rogu są góry o wysokości 1000 stóp. Wprawdzie z kabiny pilota są one niewidoczne, ale można się o nie rozbić, jeśli będziemy lecieć niżej niż na wysokości 1000 stóp. Jeżeli natomiast polecimy ponad górami na wysokość 1000–2000 stóp, to radiowy wysokościomierz będzie podawał wysokość lotu liczoną od szczytów gór, czyli o 1000 stóp mniej niż w przypadku drugiego wysokościomierza.

Jeśli chcemy sobie zapewnić warunki bezpiecznego lotu, przestrzegajmy następujących zasad:

- Nie zderzajmy się z ziemią, górami i jeziorami!
- Nie lądujmy ze schowanym podwoziem albo z szybkością pionową przekraczającą 1000 stóp/min.
- Nie kołujmy z dużą szybkością poza pasem startowym.
- Posługujmy się rozważnie klapami, opuszczając je lub podnosząc zależnie od prędkości.

Naruszenie tej ostatniej zasady prowadzi do przechyłu maszyny na jedną stronę i opadania w dół. Jeżeli dysponujemy jeszcze odpowiednim zapasem wysokości, to możemy zwiększyć "gaz" i uratować sytuację.

FIGHTER PILOT

Program symuluje lot samolotem F-15. Możliwa jest pełna akrobacja, można też ćwiczyć walkę powietrzną. Rysunek 2 przedstawia widok tablicy rozdzielczej.

● LĄDOWANIE (Landing Practice)

Samolot zbliża się od południa do głównego pasa startowego (BASE), lecąc na wysokości 1700 stóp z prędkością 6 mil. Podwozie opuszczone, gotowe do lądowania. Korzystając z informacji dostarczonych przez przyrząd kontroli lądowania (ILS) lub komputer pokładowy, można regulować wytracanie wysokości za pomocą drążka i klap. Po udanym lądowaniu należy całkowicie "zdjąć" gaz, a następnie – przy użyciu hamulców – zatrzymać samolot.

● LOT ĆWICZEBNY (Flying Training)

Samolot znajduje się na południowym skraju pasa startowego BASE, dziobem na północ. Aby wystartować, należy dodać gazu, a po osiągnięciu szybkości startowej ściągnąć drążek. Po oderwaniu się od ziemi musimy niezwłocznie, zanim osiągniemy szybkość 300 węzłów, wciągnąć podwozie. Szybkość startu możemy zmniejszyć ze 140 do 130 węzłów otwierając klapy. W trakcie kołowania możemy zmieniać kierunek posługując się sterem statecznika pionowego.

● ĆWICZENIE WALKI POWIETRZNEJ (Air to Air Combat Practice)

Samolot nasz leci na tej samej wysokości co maszyna nieprzyjacielska, z tym że znajduje się za nią, w odległości 2 mil. Po włączeniu komputera pokładowego uzyskamy informację na temat kierunku lotu, odległości i pułapu lotu samolotu nieprzyjaciela. Przeciwnik ucieka z prędkością 550 węzłów i nie odpowiada na ataki. Gdy go zobaczymy, możemy spróbować naprowadzić na niego celownik, a następnie otworzyć ogień.

● WALKA POWIETRZNA (Air to Air Combat)

Naszym zadaniem jest obrona czterech lotnisk (BASE, TANGO, DELTA i ZULU). Po starcie z BASE należy zbliżyć się do nieprzyjaciela, korzystając z radaru i komputera pokładowego. Nieprzyjaciel staje się widoczny z odległości mniejszej niż 1 mila, przy różnicy wysokości lotu mniejszej niż 5000 stóp. Uszkodzenia własne sygnalizuje zmiana barwy małego samolotu, widocznego na środku ekranu radaru. Czwarte trafienie jest ostateczne. Rozsądek nakazuje więc, by poważnie uszkodzony samolot wycofać z walki. UWAGA! – jeżeli z powodu braku amunicji lub z powodu uszkodzenia samolotu będziemy zmuszeni wycofać się z walki, nieprzyjaciel może nas trafić jeszcze jeden raz, nawet jeśli będziemy oddaleni bardziej niż o milę, a różnica wysokości będzie większa niż 5000 stóp. W takim przypadku nieprzyjacielski samolot kontynuuje lot w kierunku upatrzonego lotniska. Dzięki temu zyskujemy na czasie i możemy lądować na którymś z lotnisk w celu dokonania niezbędnych napraw względnie uzupełnienia zapasu paliwa.

● LĄDOWANIE "NA ŚLEPO" (Blind Landing)

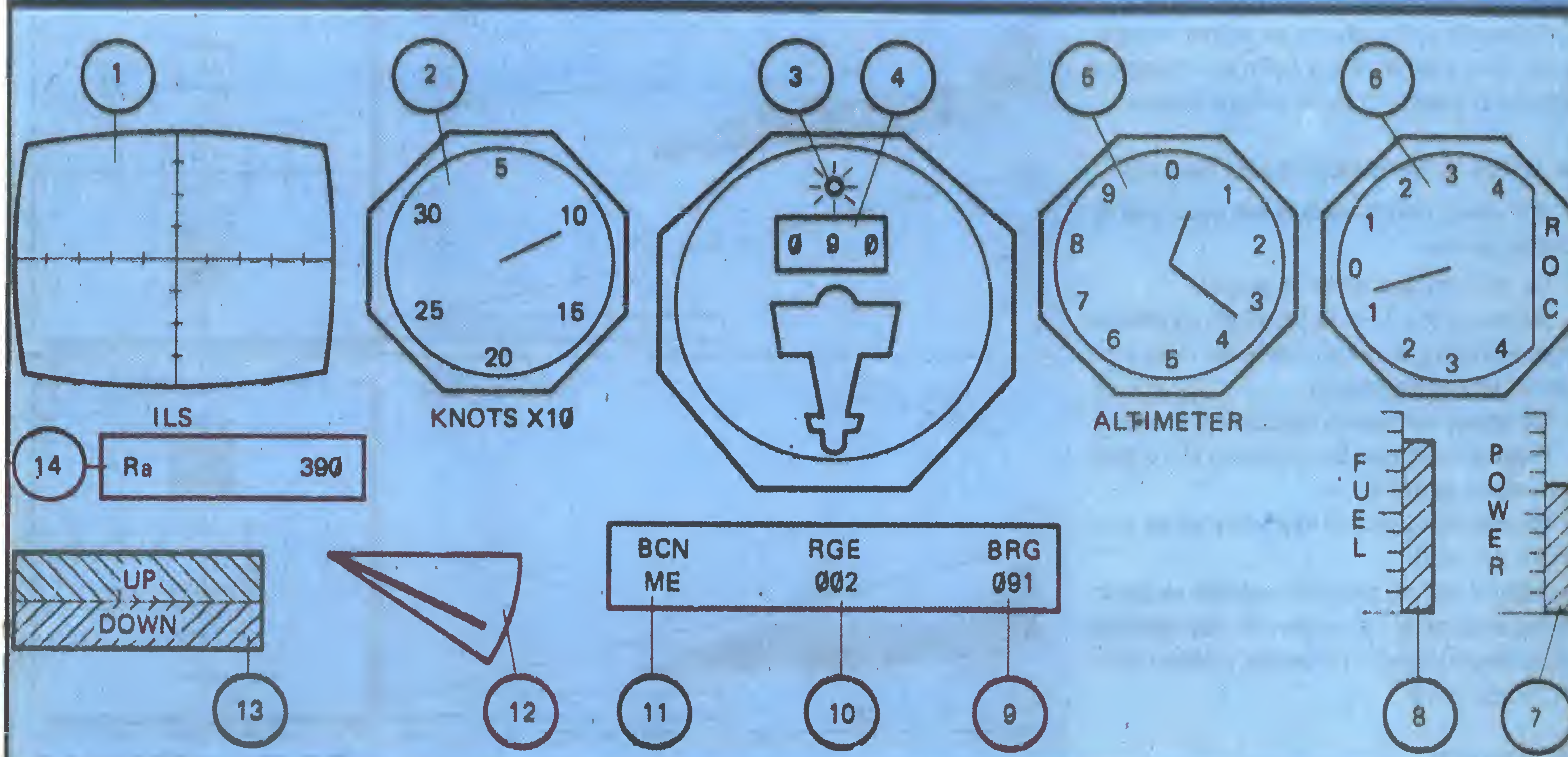
Program imituje mglistą pogodę. Horyzont jest całkowicie niewidoczny, zaś ziemię widzimy z wysokości mniejszej niż 50 stóp. Nawigacja możliwa jest wyłącznie za pomocą mapy i przyrządów kontrolno-pomiarowych.

● WIATR (Crosswind and Turbulence)

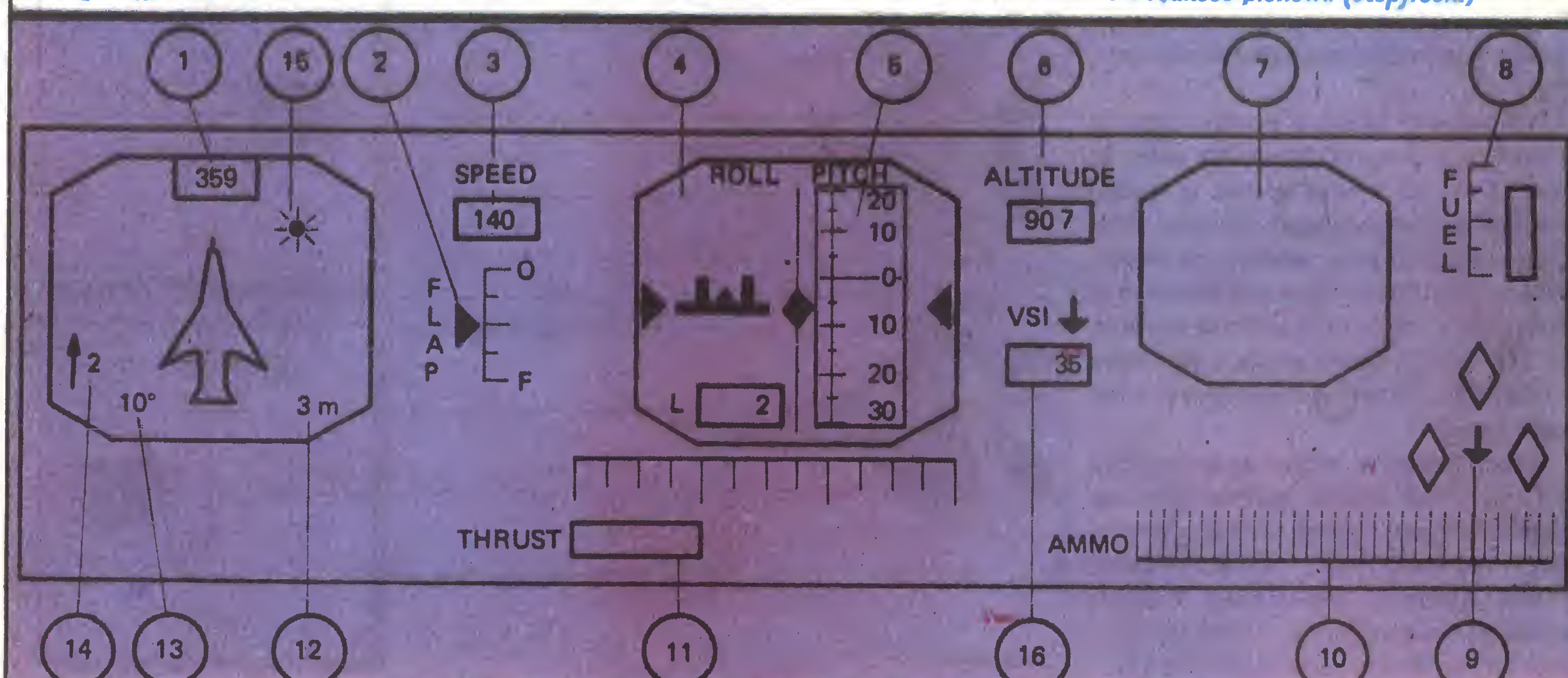
Przypadkowe, niespodziewane podmuchy wiatru sprawiają, iż samolot jest znoszony, zbacza z kursu.

▶ 30

Rys. 1 FLIGHT SIMULATOR – tablica przyrządów
1. Przyrząd lądowania ILS
2. Wskaźnik prędkości w węzłach x 10 (1 węzeł   1,85 km/h)
3. Pozycja radiostacji w stosunku do samolotu (migający punkt)
4. Kierunek lotu w stopniach
5. Wskaźnik wysokości
Mała wskazówka: 1000 stop
Duża wskazówka: 100 stop (1 stopa 0,3 m)
6. „Wariometr” do pomiaru szybkości pionowej (wyskalowanie: 1000 stóp/min): do góry: wznoszenie, do dołu: opadanie
7. Ciąg silnika
8. Wskaźnik poziomu paliwa
9. Kierunek w stopniach
10. Odległość w milach (1 mila   1,85 km)
11. Nazwa namierzanej radiostacji
12. Klapy: zamknięte,   otwarte
13. Podwozie: wciągnięte,   opuszczone
14. Radiowy pomiar wysokości (mierzy tylko poniżej 1000 stop)



Rys. 2 FIGHTER PILOT – tablica przyrządów
1. Kierunek lotu w stopniach
2. Klapy ↑ zamknięte ↓ otwarte
3. Prędkość w węzłach
4. Przechył w stopniach
L: na lewo (left), R: na prawo (right)
5. Natarcie w stopniach
niebieski (ciemny): do góry, żółty (jasny): do dołu
6. Wysokość w stopach
7. Przyrząd lądowania ILS albo komputer pokładowy
8. Wskaźnik poziomu paliwa
9. Podwozie: ↓ (zielony) opuszczone, ↑ (czerwony) wciągnięte
10. Ilość amunicji
11. Ciąg silnika
12. Odległość w milach
13. Kierunek radiostacji lub nieprzyjacielskiego samolotu w stopniach
14. Sygnał wywoławczy namierzanej radiostacji
15. Pozycja radiostacji lub nieprzyjacielskiego samolotu w stosunku do maszyny (migający punkt)
16. Prędkość pionowa (stopy/sek.)



Prawidłowy pilotaż w tych warunkach wymaga treningu i pewnej praktyki.

● SPRAWNOŚĆ PRZECIWNIKA (Pilot Rating)

Sprawność przeciwnika może być zmieniana – od "żółtodzioba" po "asa". Zmiana tego typu określa jak szybko zauważymy zbliżającego się nieprzyjaciela, jak szybko weźmie on na celownik nasz samolot czy też z jakiej odległości można zadać decydujący strzał.

Sprawność przeciwnika nie ma wpływu na możliwości naszej maszyny. Warto zachować ostrożność w potyczce z "asem" – to bardzo trudny przeciwnik.

● WYBÓR ODPOWIEDNIEGO DRĄŻKA (Controls)

Oto zasady, których należy przestrzegać, żeby lecieć bezpiecznie:

– Nie zderzajmy się z ziemią lub górami!
– Nie lądujmy ze schowanym podwoziem z prędkością pionową większą niż 10 stóp/minutę albo mając przechył większy niż 2-3 stopnie!
– Nie kołujmy z prędkością większą niż 250 węzłów!
– Podwozie nie powinno być opuszczone przy prędkości większej niż 300 węzłów!
– Nie zmieniajmy położenia klap, jeśli prędkość przewyższa 472 węzły!
– Szybkość lotu przy zamkniętych klapach nie powinna być mniejsza niż 130 węzłów, zaś przy całkowicie opuszczonych klapach – 120 węzłów, ponieważ samolot "przekoziołkuje"!

● NAWIGACJA


W nawigacji lotniczej kierunek podawany jest w stopniach. Kierunek 0° oznacza północ, 90° – wschód, 180° – południe i 270° – zachód.

Z takim właśnie namiarem położenia mamy do czynienia w przypadku przyrządów określających kierunek lotu (kompas) oraz w przypadku radiowych wskaźników kierunku, które podają usytuowanie pasa startowego.

Jeśli chcemy lecieć w kierunku radiostacji, musimy się obracać tak długo, aż kierunek lotu będzie się zgadzał z kierunkiem radiostacji. Zadanie to okaże się o wiele trudniejsze, jeśli będziemy chcieli wybrać kierunek, z jakiego dolecimy do radiostacji (chcemy np. wylądować, co jest możliwe tylko wzdłuż osi pasa startowego).

Przy lądowaniu najbardziej przydatny okaże się przyrząd ILS. Informuje on na ile tor lotu odbiega od ideału. Na rysunku 3 widać wyraźnie, że migający punkt zmienia położenie zgodnie z kierunkiem, w którym powinniśmy skierować samolot w celu skorygowania toru lotu. UWAGA! – im bardziej zbliżamy się do lotniska, tym przyrząd staje się czulszy na odchylenia. Nie zmniejszajmy gwałtownie wysokości tylko dlatego, że mamy trochę "zapasu", czyli znajdujemy się trochę wyżej niż trzeba.

Komputer pokładowy umożliwia bardzo precyzyjne określenie pozycji samolotu, ponieważ sygnalizuje odległość i pozycję w stopniach z podziałem na strony świata (N=North=Północ, S=South=Południe, E=East=Wschód, W=West=Zachód). Metodą tą możemy się posługiwać jedynie wówczas, jeśli radiostacja znajduje się w odległości mniejszej niż 6 mil.



FLIGHT SIMULATION – KLAWISZE STERUJĄCE

		↓ (6) Drążek do siebie (wznoszenie)
↑ (7) Drążek od siebie (opadanie)	→ (8) Skręt w prawo (przechył na prawo)	← (5) Skręt w lewo (przechył w lewo)
Z Orczyk w lewo (zwrot w lewo)	X Orczyk w prawo (zwrot w prawo)	P Dodanie gazu (większy ciąg)
C Cofnięcie gazu (mniejszy ciąg)	F Otwarcie klap	D Zamknięcie klap
G Wciągnięcie opuszczenie podwozia	B Wybór radiostacji	M Przełączenie na mapę

FIGHTER PILOT – KLAWISZE STERUJĄCE

		↓ (6) Drążek do siebie (wznoszenie)
↑ (7) Drążek od siebie (opadanie)	→ (8) Skręt w prawo (przechył na prawo)	← (5) Skręt w lewo (przechył w lewo)
CShift Orczyk w lewo (zwrot w lewo)	Z Orczyk w prawo (zwrot w prawo)	Q Dodanie gazu (większy ciąg)
A Cofnięcie gazu (mniejszy ciąg)	S Otwarcie klap	W Zamknięcie klap
U Wciągnięcie opuszczenie podwozia	N Wybór radiostacji	M Przełączenie na mapę
B Hamulce	SShift Przełącznik przyrządu ILS, komputera pokład.	C Przełączenie na bojowy styl lotu
O OGIEŃ (w bojowym stylu lotu)	H Pauza	J Ponowne uruchomienie programu

● MECHANIKA LOTU

W czasie lotu trzeba pokonać przede wszystkim grawitację – samolot musi oddalać się od powierzchni ziemi. Jest to możliwe dzięki sile nośnej powstającej na płacie nośnym, czyli skrzydle, której wielkość zależy od prędkości lotu i od kształtu skrzydła. Jeśli np. w locie poziomym prędkość ulegnie zmniejszeniu, to zmniejszy się także siła nośna, w wyniku czego samolot będzie tracił wysokość. Klapy umożliwiają zmianę kształtu skrzydeł. Jeśli je opuścimy, to przy tej samej prędkości siła nośna będzie większa, możemy więc pozostać w powietrzu lecąc z mniejszą prędkością. Tę właściwość klap wykorzystujemy przy lądowaniu, kiedy nie chcemy kołować ze zbyt dużą prędkością.

Podobnie rzecz się ma z oporem powietrza. Jego wielkość zależy od szybkości, od kształtu samolotu i od gęstości powietrza. Uzyskanie dużej szybkości wiąże się z pokonaniem dużego oporu powietrza. Aby było to możliwe, trzeba dysponować silnikiem o odpowiednim ciągu. UWAGA! – w programie "FIGHTER PILOT" przy bardzo małych szybkościach następują silniejsze zawirowania powietrza wokół skrzydeł, które w efekcie zwiększają opór powietrza i powodują spadek ciągu silnika. Można temu przeciwdziałać jedynie poprzez zwiększenie gazu. Oczywiście wiąże się to z przyspieszeniem; później należy zwolnić i tak dalej. Tych manewrów przyspieszaczem można uniknąć opuszczając klapy podczas zmniejszania prędkości (300-400 węzłów). Na zmianę kształtu samolotu, oprócz klap, wpływa także podwozie. Jeśli jest opuszczone, opór powietrza jest większy.

Mniejsza gęstość powietrza na dużych wysokościach powoduje, że opór maleje. W programie "FIGHTER PILOT" np. na niewielkich wysokościach można latać z prędkością maksymalną 800 węzłów, natomiast na wysokości 60 tys. stóp prędkość maksymalna wynosi 1440 węzłów.

Siłę ciągu, konieczną do pokonania oporu powietrza i do wznoszenia się, zapewnia silnik. Ponieważ do spalania mieszanki wykorzystywany jest tlen zawarty w powietrzu, w wyższych partiach atmosfery, gdzie powietrze jest rzadsze, wydajność silnika maleje. Samoloty myśliwskie w większości wyposażone są w dopalacze. Urządzenie to włącza się przy całkowitym i ciągłym wciskaniu dźwigni przyspieszenia. Dopalacz wtryskuje do komory spalania dodatkową ilość paliwa, dzięki czemu wzrasta szybkość, ale oczywiście i zużycie paliwa. W programie "FIGHTER PILOT" trzecia (czerwona) część skali przyrządu uciągu sygnalizuje działanie dopalacza.

Kierunek lotu możemy zmienić wykonując odpowiedni zwrot – wystarczy przechylić samolot w pożądanym kierunku, przy czym promień skrętu będzie tym mniejszy, im bardziej przechylimy maszynę. Przy zwrotach, wykonywanych ze znacznym przechyłem, dostrzeżemy kilka ciekawych zjawisk. Siła odśrodkowa (w poziomie) nakłada się na siłę przyciągania (masę samolotu) i maszyna gwałtownie traci wysokość, opada w dół.

W ten sposób przy dużym przechyle (powyżej 45°) można zauważyć zmianę funkcji sterów: orczykiem, a więc sterem statecznika pionowego można regulować kąt nurkowania lub wznoszenia się, zaś drążkiem, a więc sterem statecznika poziomego można "zwęzić" przechył.

Opr. SZCZEPAN WORONOWICZ

Na podstawie Gyorgy Háy, Repulesszimulációk pilótaszemmel, Nikroszámitógep Magazin, nr 3/1985

GŁOS PANA ALBO O PRZERWANIACH W ZX SPECTRUM

Komputer musi często wykonywać w pewnej jednostce czasu wiele różnych czynności. Najprościej użyć kilku procesorów, z których każdy odpowiadałby za jedno zadanie. Na ogół jednak zadania te wymagają tylko krótkotrwałej aktywności – w pozostałym czasie procesory byłyby nie wykorzystane.

"Baron de Break utrzymuje liczną świtę. Ogrodnik Jan, masztalerz, kamerdyner – każdy pełni inną funkcję. Zwykle pracuje jeden, zaś inni się wałkonią".

Rozwiązaniem bardziej ekonomicznym jest zastosowanie jednego procesora, spełniającego różne funkcje. Umożliwia to mechanizm przerwań, pozwalający realizować w tym samym czasie kilka niezależnych zadań. Czynności nie cierpiące zwłoki wykonywane są natychmiast, na sygnał z zewnątrz. Po ich zakończeniu procesor może powrócić do czynności żmudnych, lecz mniej pilnych.

"Odkąd de Break zwolnił resztę służby, Janowi, ma się ku gorszemu. Zwykle pielęgnuje klomby i gazony zamkowego ogrodu. «Janie!!!» rozlega się jednak raz po raz gromki głos Pana. Jan rzuca szpadel i pędzi na wezwanie. Upudruje perukę, osiodła dżaneta, zawiąże halsztuk, po czym wróci do ogrodu, podniesie porzucone narzędzia i podejmie pracę w tym samym miejscu, w którym przerwało ją wołanie barona".

Przerwanie polega na zawieszeniu, "uśpieniu" realizowanego programu na zewnętrzny sygnał i zatrudnieniu procesora na pewien czas do pilniejszych prac. Przerwanie przypomina nieco wywołanie podprogramu maszynowego, z tym że podprogram wywoływany jest nie na życzenie programu, lecz na wyraźne żądanie z zewnątrz komputera, wyrażone zamianą poziomu logicznego na odpowiedniej końcówce. Tak więc podczas gdy "zwykły" podprogram "świadczy usługi" na rzecz programu, który go wywołał, otrzymuje od niego i przekazuje mu parametry, to program obsługi przerwania nie ma nic wspólnego z przerwanym programem. Po zakończeniu obsługi przerwania stan procesora musi być identyczny, jak w chwili przyjęcia przerwania. Po "przebudzeniu z narkozy" przerwany program nie może "zauważyć", że w międzyczasie coś się zmieniło, np. zawartość rejestrów lub bity stanu.

Przerwania wykorzystuje, i to 50 razy na sekundę, nawet tak prosty mikrokomputer, jak ZX Spectrum. W regularnych odstępach czasu układ scalony ULA wysyła sygnał przerwania, uruchamiając procedurę obsługi, zawartą w pamięci ROM. Jej zadaniem jest testowanie klawiatury i odmierzanie czasu przez inkrementację (zwiększanie o 1) zmiennej systemowej FRAMES (23672-4).

Zastosowany w ZX Spectrum procesor Z80A ma dwie końcówki przerwań: NMI i INT. Pierwsza wywołuje tzw. przerwanie niemaskowalne, obsługiwane natychmiast i w każdych warunkach. Ponieważ jego wykorzystanie w Spectrum bez przeróbek sprzętowych jest niemożliwe, nie będziemy się nim zajmować. Przerwanie INT to tzw. przerwanie maskowalne. Jeśli w danej chwili przerwanie jest mikroprocesorowi "nie na rękę", może go uniknąć. Specjalny rozkaz maszynowy DI sprawia, że procesor nie reaguje na sygnały końcówki INT aż do chwili napotkania rozkazu EI, ponownie zezwalającego na przerwanie programu.

Z80 ma trzy tzw. tryby obsługi przerwania INT, ustawiane rozkazami IM 0, IM 1, IM 2. Trybu 0 w Spectrum użyć nie można. Normalnie używany jest najprostszy tryb 1. Po przyjęciu sygnału przerwania następuje wywołanie podprogramu pod adresem 56, analogicznie jak po CALL 56 lub RST 7. Pod tym adresem w pamięci stałej (ROM) znajduje się skok do właściwej procedury obsługi, umieszczonej w dalszej części ROM. W trybie 1 tok postępowania po przerwaniu jest sztywno ustalony i nie daje nam szans ingerencji.

Aby użyć przerwań do własnych celów, przełączymy procesor w tryb 2, zwany wektoryzowanym. Procedur obsługi może być wiele (do 128) i mogą one być dowolnie rozmieszczone w pamięci. Ich adresy są zebrane w specjalnej tablicy. Starszy bajt adresu tej tablicy należy wpisać do rejestru I procesora. Po przyjęciu przerwania procesor oczekuje na tzw. wektor adresu przerwania. Normalnie wysyła je urządzenie, które przerwanie wywołało. Wektor ten jest młodszym bajtem adresu tego elementu tablicy, który zawiera adres właściwej procedury obsługi przerwania. W ten sposób każde z urządzeń wywołujących przerwanie INT może pośrednio wskazać własną procedurę obsługi. Oto przykład: rejestr I zawiera 16. Po przerwaniu procesor odczytał z magistrali danych wektor 4. Adres adresu wynosi więc 256 x 16 + 4 = 4100. Pod tym adresem mamy słowo o wartości 64000, które jest właściwym adresem procedury obsługi przerwania. Starszy bajt adresu wszystkich elementów tablicy musi być taki sam. W przypadku procesorów 8-bitowych obszar pamięci, w którym wszystkie adresy mają identyczny starszy bajt, nazywamy stronicą. Stronica liczy więc 256 bajtów i może pomieścić 128 dwubajtowych adresów. To samo dotyczy tablicy adresów przerwań, gdyż jej rozmiar jest ograniczony do jednej strony.

W ZX Spectrum nie ma urządzeń zewnętrznych, dostosowanych do pracy w trybie 2. Po przełączeniu w tryb 2 procesor "w dobrej wierze" odczyta jednak stan magistrali danych i przyjmie go za młodszy bajt adresu wektora, wysłany przez urządzenie. Jeśli nie dołączyliśmy do Spectrum żadnych urządzeń zewnętrznych, procesor odczyta same jedynki (=255), gdyż w takim stanie jest magistrala danych w stanie spoczynku. Gdy do złącza krawędziowego przyłączona jest np. drukarka lub interfejsy, stan magistrali może być przypadkowy, niemożliwy do przewidzenia. Co robić?

Trzeba uniezależnić wyznaczony przez procesor adres procedury obsługi od wektora odczytanego z magistrali. Gdyby tak wypełnić całą tablicę tą samą zawartością, to procesor za każdym razem pobierałby ten sam adres, choć być może z różnych komórek tablicy. Proste? Tak, ale... Dokumentacja procesora żąda, żeby wektor był liczbą parzystą, czyli miał wyzerowany najmłodszy bit (bit nr 0). W naszym przypadku nie możemy tego zagwarantować, ba, przy odczytanym "wektorze" 255 bit nr 0 na pewno będzie miał wartość 1. Jak wtedy zareaguje procesor? Dane producentów tego nie precyzują.

W praktyce procesory stosowane w ZX Spectrum akceptują zarówno wektory parzyste, jak i nieparzyste. Niech rejestr I zawiera 1. Wektor 0 spowoduje odczytanie adresu z komórki 256 (młodszy bajt) i 257 (starszy), zaś wektor 1 z komórek 257 i 258. Ten sam bajt tablicy może być raz interpretowany jako starszy, innym razem jako młodszy bajt adresu!

Jedyne wyjście to umieścić procedurę obsługi przerwania pod takim adresem, którego młodszy i starszy bajt są identyczne, np. 11110000 11110000, czyli szesnastkowo F0F0 (dziesiętnie: 61680). Nadają się oczywiście i inne adresy o podobnych właściwościach: F7F7 (hex) itd. Całą stronicę pamięci, przeznaczoną dla tablicy, wypełniamy identyczną zawartością, np. F7H (hex). Taką samą wartość należy też wpisać do pierwszej komórki stronicy następnej, a to na wypadek wektora o wartości 255.

Przystępujemy do działania. Instrukcją np. CLEAR 63199 rezerwujemy obszar pamięci od komórki 63200 włącznie, chroniąc go przed przypadkową ingerencją interpretera języka BASIC. Upatrujemy adres procedury obsługi np. F8F8H (F8H = 248; 256x248+248=63736). Wybieramy stronicę dla tablicy adresów (np. stronica nr 247, adres: 63232) i 257 kolejnych bajtów wypełniamy jedną wartością 248. Te czynności możemy wykonać w języku BASIC, jednak dalsze operacje trzeba będzie zaprogramować w asemblerze.

Zacznijmy od procedury obsługi przerwania. Niech wykona ona prostą czynność, polegającą na zapisywaniu stałą wartością pierwszych 256 bajtów pamięci atrybutów. Pozwoli to utworzyć w górnej tercji ekranu okno o ustalonych barwach wypełnienia i tła (np. czerwone na żółtym) i podwyższonej jaskrawości. Pierwszą i obowiązkową czynnością programu obsługi musi być przechowanie na stosie zawartości akumulatora i bitów stanu (PUSH AF) oraz wszystkich reje-

strów wykorzystywanych przez podprogram. Następnie ładujemy rejestr HL adresem pierwszej komórki pamięci atrybutów, zaś rejestr B zerujemy. Pętla wpisuje stałą 114 do kolejnych komórek pamięci. Gdy po 256 powtórzeniach rejestr B ponownie się wyzeruje, program odtworzy rejestry i "skoczy" do normalnej procedury obsługi przerwania w pamięci ROM – jak po zwykłym przerwaniu w trybie 1. Dla lepszej czytelności kod podano w systemie szesnastkowym, lecz adresy – w dziesiętnym:

```
63736 F5          PRZER: PUSH AF        ;przechowaj na stosie
63737 E5                 PUSH HL        ;rejestry: AF, HL, BC
63738 C5                 PUSH BC        ;
63739 21 00 58           LD HL,22528    ;HL=adres pam. atryb.
63742 06 00              LD B,0         ;B = licznik (0 =256)
63744 36 72       PETLA: LD (HL),114    ;wpisz 114  pod adres
63746 23                 INC HL         ;z HL, zwiększ HL o 1
63747 10 FB              DJNZ PETLA     ;zmniejsz B o 1,jeśli
63749 C1                 POP BC         ;nie 0, powtórz pętlę
63750 E1                 POP HL         ;odtwórz    zawartość
63751 F1                 POP AF         ;rejstrów BC, HL i AF
63752 C3 38 00           JP 56          ;skocz do prog. w ROM
```

Przełączenia trybów reakcji też musi dokonac program maszynowy. Wpierw trzeba wpisać do rejestru wektora przerwań I nr strony z tablicą adresów (niezbędne jest pośrednictwo akumulatora), a potem można włączyć tryb 2 rozkazem IM 2. Powrót do starego sposobu obsługi jest jeszcze prostszy: wystarczy spowodować wykonanie rozkazu IM1:

```
63755 3E F7       WLACZ: LD A,247       ;wpisz do A nr strony
63757 ED 47              LD I,A         ;przepisz zaw. A do I
63759 ED 5E              IM 2           ;włącz tryb IM 2
63761 C9                 RET            ;wróć do języka BASIC

63762 ED 56       KASUJ: IM 1           ;przywróć tryb IM 1
63764 C9                 RET            ;wróć do języka BASIC
```

Oto program ładujący do pamięci kod maszynowy przedstawionych programów. Po pomyślnym zakończeniu ładowania instrukcją RAND USR 63755 inicjujemy nowy tryb obsługi i od tej chwili po każdym przerwaniu wywoływana jest procedura PRZER. Górna tercja ekranu pozostaje niewrażliwa na wszelkie zmiany atrybutów z poziomu języka BASIC:

```
 10 CLEAR 63199
 20 LET SU=0: LET T=63232: LET B=248
 30 FOR I=0 TO 256: POKE T+I,B: NEXT I
 40 FOR A=63736 TO 63764
 50 READ BY: POKE A,BY: LET SU=SU+BY
 60 NEXT A
 70 IF SU<>3851 THEN  PRINT "BLAD": STOP
 80 RANDOMIZE USR 63755
 90 DATA 245,229,197,33,0,88,6,0,54,114
100 DATA 35,16,251,193,225,241,195,56,0
110 DATA 62,247,237,71,237,94,201,237
120 DATA 86,201
```

Procedura obsługi pełni swą powinność "za plecami" programu głównego, czyli interpretera języka BASIC. Jej czujną obecność łatwo sprawdzić, zmieniając stałą w rozkazie LD (HL),114. Stała stanowi drugi bajt rozkazu i znajduje się w komórce 63745:

POKE 63745,102

Kolory górnej części ekranu ulegają natychmiastowej zmianie. Jeśli wyłączymy procedurę przez RAND USR 63762, wszystko wróci do normy.

Nie przewidując współpracy Spectrum z przystawkami dołączanymi do złącza krawędziowego, można zrezygnować z 255 początkowych bajtów tablicy adresów, pozostawiając tylko dwa ostatnie. Przekona nas o tym zlecenie:

FOR I= 0 TO 254: POKE T+I,0: NEXT I

Wartość wektora wynosi za każdym razem 255 i komputer odczytuje adres z komórek 65487 i 65488. W tym przypadku po wyzerowaniu reszty komórek nic się nie zmieni, a do komórek 65487/8 można wpisać dowolny adres.

Przerwania pozwalają w prosty sposób rozwiązać problemy trudne do zaatakowania w innych sposób. Przykład: podczas listowania programu Spectrum wyświetla stronę po stronie, za każdym razem pytając "scroll?". Często byłoby wygodniej przeglądać program w sposób ciągły, zatrzymując listing w dowolnej chwili naciśnięciem np. kombinacji klawiszy (Z) i (X). Po zwolnieniu któregoś z przycisków listing byłby kontynuowany:

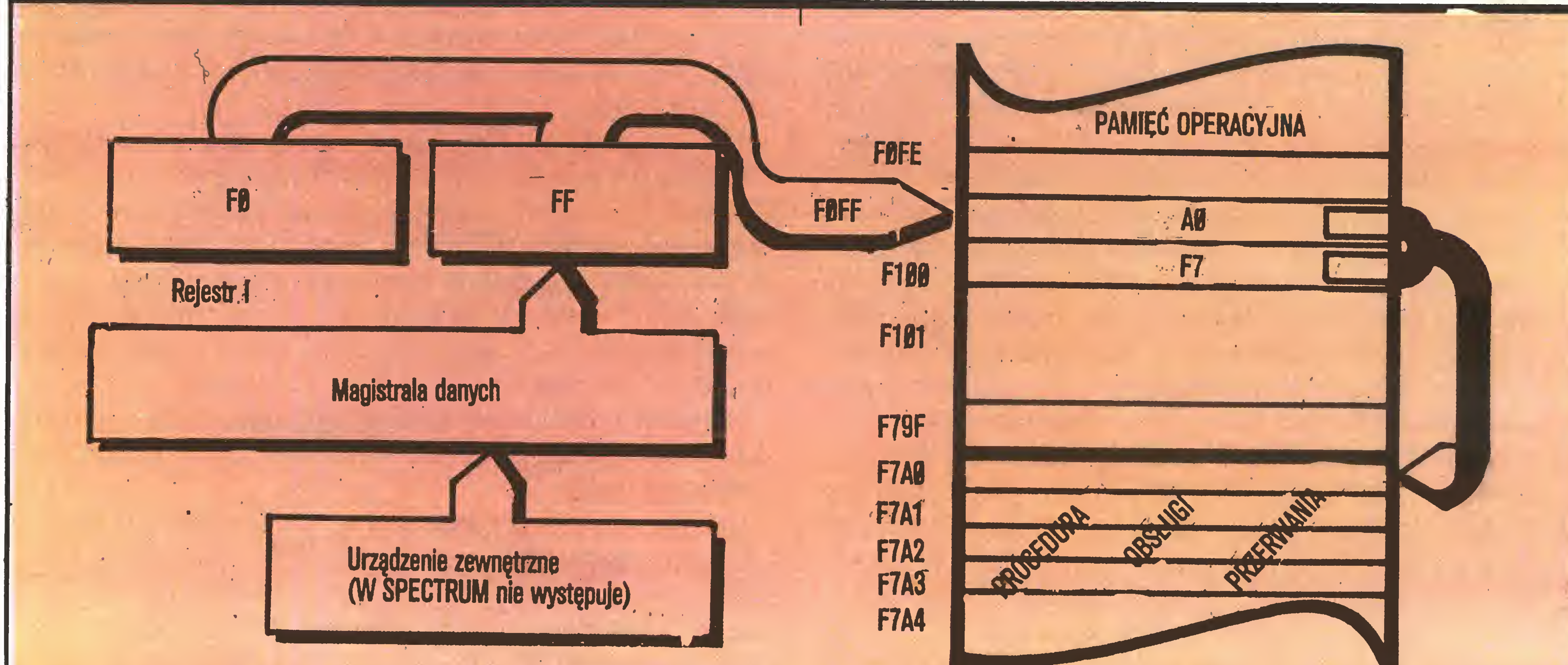
```
63736 F5          NSTOP: PUSH AF          ;przechowaj rejestry AF
63737 3E FF              LD A, 255        ;do komórki 23692 wpisz
63739 32 8C 5C           LD (23692),A     ;255, wyłączając scroll
63742 3E FE        TEST: LD A, 254        ;wybierz grupę klawiszy
63744 DB FE              IN A, (254)      ;odczytaj    klawiaturę
63746 FE B9              CP 185           ;stan klawiatury = 185?
63748 28 F6              JR Z, TEST       ;gdy tak,ponowić odczyt
63750 F1                 POP AF           ;odtwórz  rejestry   AF
63751 C3 38 00           JP 56            ;skocz do obsługi w ROM
```

Po każdym przerwaniu procedura uniemożliwia ewentualny scrolling, nadając zmiennej systemowej SCRCT (adres: 23692) wartość 255. Następnie odbywa się testowanie klawiatury. Para rozkazów: LD A,254 i IN A,(254) jest funkcjonalnym odpowiednikiem instrukcji LET A= IN 65276 w języku BASIC. Jeśli po IN A,(254) w akumulatorze znajdzie się liczba 185, oznacza to równoczesne wciśnięcie (Z) i (X). W tym przypadku program oczekuje w pętli, aż któryś z tych przycisków zostanie zwolniony. Ponieważ program zajmuje ten sam obszar pamięci co poprzednia procedura i korzysta z identycznych podprogramów WLACZ i KASUJ, w programie ładującym wystarczy zmienić trzy instrukcje. Procedurę można wyłaczyć przez RAND USR 63762:

```
 70 IF SU<>4525 THEN  PRINT "BLAD": STOP
 90 DATA 245,62,255,50,140,92,62,254,219
100 DATA 254,254,185,40,248,241,195,56,0,0
```

Przerwania pozwalają uzyskać wspaniałe efekty (podkład muzyczny, animacja ekranu itd.), które podziwiamy w najlepszych grach. Niektóre mikrokomputery np. C64/128 – mają znacznie szersze niż Spectrum możliwości korzystania z przerwań. Tak czy owak, mechanizm przerwań jest potężnym narzędziem w ręku programisty, a technika jego stosowania nie jest wiedzą tajemną, zastrzeżoną dla profesjonalistów. Prawda, programowanie obsługi przerwań wymaga szczególnej staranności, gdyż pracę programu trudno jest testować "na sucho", krok po kroku, korzystając np. z monitora kodu maszynowego. Za każdy błąd Spectrum ochoczo się rewanżuje, informując nas, kto i kiedy skonstruował nasz mikrokomputer...

ROLAND WACŁAWEK



Jak mikroprocesor Z80 wyznacza adres procedury obsługi przerwania w trybie ZM2.



O SKUTECZNYM SORTOWANIU [1]

Sortowanie jest zadaniem spotykanym w codziennym życiu niemal na każdym kroku. Powszechnie mamy do czynienia ze zbiorami obiektów uporządkowanymi według pewnych kryteriów. Trudno przecież sobie wyobrazić użyteczną książkę telefoniczną z ustawionymi przypadkowo nazwiskami abonentów, słownik, w którym hasła są umieszczone w sposób przypadkowy czy bibliotekę z książkami wstawionymi na półki "jak leci". Wszędzie tam, gdzie trzeba przeszukiwać zbiór obiektów w celu znalezienia jednego (kilku) potrzebnego, wygodnie jest wcześniej uporządkować, czyli posortować dany zbiór według określonego klucza. Dla książek w bibliotece takim kluczem będzie numer ewidencyjny książki, w słowniku języka polskiego kluczem będzie słowo, którego dotyczy treść hasła, a wprowadzony porządek będzie porządkiem alfabetycznym. Uprzednie posortowanie zbioru pozwala w łatwy sposób tworzyć od razu uporządkowane zestawienia obiektów spełniających pewne warunki. Można na przykład szybko utworzyć z książki telefonicznej małą książkę abonentów z pewnego rejonu – tych, których numer telefonu zaczyna się np. na 27.

Mówiąc ściśle, sortowanie ciągu polega na takiej zmianie kolejności jego elementów, aby w wyniku otrzymać ciąg niemalejący (ustawienie od najmniejszego do największego) lub nierosnący (ustawienie od największego do najmniejszego). Jedyne wymaganie stawiane sortowanym elementom to możliwość ich porównywania tj. określania, który obiekt jest mniejszy (powinien wystąpić w ciągu wcześniej) a który większy (powinien wystąpić później przy porządku niemalejącym). Gdy klucze, według których sortujemy, są liczbami lub słowami pewnego alfabetu, możliwość określania ich kolejności jest oczywista. Czasem można i warto dodatkowo skorzystać z innych własności porządkowanych elementów.

Algorytmy sortowania można ogólnie podzielić na dwie klasy. Pierwszą stanowią algorytmy wykorzystujące specyficzne własności obiektów. Do drugiej zalicza się te, które zadowalają się jedynie możliwością porównywania elementów.

Na pierwszy rzut oka wydaje się, że problem sortowania nie jest wcale złożony. Można natychmiast podać łatwe i naturalne metody jego rozwiązania. Tymczasem okazuje się, że te proste metody są dobre i skuteczne jedynie wtedy, gdy mamy do czynienia z niewielką liczbą obiektów do uporządkowania. Jeśli tych obiektów jest bardzo dużo, zastosowanie jednego ze skomplikowanych, lecz efektywnych algorytmów przynosi ogromne korzyści.

Sortowanie dużego zbioru danych wiąże się ze znacznym kosztem. W przypadku porządkowania ręcznego jest to znaczny nakład pracy. Gdy pracę tę wykonuje za człowieka maszyna, ważna jest liczba dokonywanych operacji – porównań elementów. Istotnego znaczenia nabiera też czas wykonywania algorytmu. Wielkość ta jest zawsze zależna od liczby porządkowanych elementów. Dobre metody sortowania pozwalają uporządkować n elementów przy pomocy liczby porównań rzędu $n\log_2 n$ (przy założeniu, że maszyna korzysta z nie więcej niż jednego procesora). Okazuje się zarazem, że rząd ten w ogólnym przypadku nie może być mniejszy. Im bardziej wyrafinowany jest algorytm sortujący, tym mniejsza jest stała stojąca przed wartością $n\log_2 n$. Jedynie algorytmy wykorzystujące specyficzne cechy sortowanych obiektów działają w czasie liniowym – rzędu liczby danych.

Wybrana metoda sortowania jest zawsze w istotny sposób zależna od struktury przetwarzanych danych. Jeśli wszystkie dane mieszczą się na raz w pamięciu wewnętrznej komputera, programista z pewnością wybierze jedną z metod sortowania wewnętrznego. Jeśli obiektów jest bardzo dużo i są one zapisane na taśmach magnetycznych, dostęp do tak zapamiętanych danych jest sekwencyjny (najpierw pierwszy obiekt, potem drugi itd.), co nakazuje wybór algorytmu z grupy metod sortowania zewnętrznego. Wszystkie rozważane poniżej algorytmy należą do pierwszej grupy – porządkują dane w pamięci wewnętrznej.

Na początek zagadnienie porządkowania ciągów o specjalnych własnościach. Zadanie: posortować n liczb całkowitych z przedziału od 1 do m. Można to zrobić w następujący sposób:

1) zainicjalizujmy m pustych kolejek, po jednej dla każdej liczby od 1 do m;
2) rozważmy wszystkie liczby w ciągu od pierwszej do n-tej wstawiając je do odpowiednich kolejek (liczba i ustawi się w kolejce o numerze i);
3) połączmy (skonkatenujmy jak mówią informatycy) wszystkie kolejki – drugą kolejkę należy dołączyć na koniec pierwszej i tak dalej aż do m-tej.

Otrzymany ciąg jest uporządkowany niemalejąco. Metoda ta, zwana algoryt-

mem sortowania rozrzutowego (ang. bucket sort) wykorzystuje fakt, że sortowane są tylko liczby całkowite z określonego przedziału. Na przykład dla liczb rzeczywistych, choćby z tego samego przedziału, zupełnie się nie nadaje – jak bowiem utworzyć kolejkę dla każdej liczby rzeczywistej z podanego zakresu?

Czas działania algorytmu jest proporcjonalny do liczby porządkowanych danych (a więc do n).

Zapis sortowania rozrzutowego w postaci programu w polskim Logo przedstawia się następująco:

```
OTO SORTOWANIE :liczby :m
INICJALIZACJA :m
ROZRZUT :liczby
WYNIK POłąCZONE.KOLEJKI :m
JUż

OTO INICJALIZACJA :m
JEŚLI :m = 0 [stop]
PRZYPISZ SłOWO "k :m []
INICJALIZACJA :m - 1
JUż

OTO ROZRZUT :liczby
JEŚLI PUSTE? :liczby [stop]
PRZYPISZ SłOWO "k PIERW :liczby
NAK PIERW :liczby WARTOść SłOWO
"k PIERW :liczby
ROZRZUT BEZPIERW :liczby
JUż

OTO POłąCZONE.KOLEJKI :i
JEŚLI :i = 0 [wynik []]
WYNIK ZDANIE POłąCZONE.KOLEJKI :
i - 1 WARTOść SłOWO "k :i
JUż
```

Liczby całkowite z podanego zakresu można szybko uporządkować w mniej wymyślny sposób (np. zamiast używać kolejek posłużyć się licznikami), ale przedstawiona metoda ma wartość dydaktyczną, gdyż powyższy algorytm można uogólnić tak, by porządkował bardziej złożone obiekty. Nie wdając się w formalne rozważania ustalmy, że teraz problem będzie polegał na sortowaniu słów. Posługując się opisaną dalej metodą, można tworzyć na przykład uporządkowane alfabetycznie listy nazwisk albo słowniki dowolnych terminów.

Zadanie: uporządkować alfabetycznie ciąg słów złożonych z liter alfabetu. A oto sposób rozwiązania tego problemu:

1) Inicjalizujemy tyle kolejek, ile jest liter alfabetu plus jedną kolejkę dodatkową.
2) Znajdujemy maksymalną długość słowa w ciągu, niech to będzie n.
3) Rozrzucamy słowa do odpowiednich kolejek, najpierw według n-tej litery tzn. słowo, którego n-tą literą jest A, ustawi się w kolejce przeznaczonej dla A, jeśli jest to B, w kolejce przeznaczonej dla B i tak dalej. Gdy słowo jest krótsze i nie ma n-tej litery, wstawia się je do dodatkowej kolejki.
4) Scalamy kolejki tak, że najpierw bierzemy kolejkę dodatkową, dołączamy na jej koniec kolejkę dla A, na jej koniec dołączamy kolejkę dla B i tak aż do Z. Otrzymamy w ten sposób ciąg słów uporządkowany według ich n-tej litery.
5) Scalony ciąg ponownie rozrzucamy do kolejek (teraz znów pustych), ale o tym do której kolejki dostanie się słowo, decyduje jego n-1-sza litera. Znów scalamy kolejki i powtarzamy dalej dwa kroki algorytmu: – rozrzut i scalanie aż do chwili, gdy połączymy kolejki, w których znajdują się słowa wstawione tam zgodnie z wartością ich pierwszej litery. Otrzymany w tym kroku ciąg słów jest już uporządkowany alfabetycznie.

Najlepiej prześledzić zasadę działania algorytmu na przykładzie. Należy ustawić alfabetycznie listę imion: OLA JAN MAREK ALA KUBA ELA

Maksymalna długość słowa wynosi 5. Inicjalizujemy kolejki oznaczając je poszczególnymi literami. Dla skrótu podane zostaną tylko kolejki odpowiadające literom istotnym w danym kroku algorytmu.

1 rozrzut – według 5-tej litery:
dodatkowa: OLA JAN ALA KUBA ELA
k : MAREK
1 scalenie:
OLA JAN ALA KUBA ELA MAREK
Jest to ciąg wyjściowy dla drugiego rozrzutu.
2 rozrzut – według 4-tej litery:
dodatkowa: OLA JAN ALA ELA
a : KUBA
e : MAREK
2 scalenie:
OLA JAN ALA ELA KUBA MAREK
3 rozrzut – według 3-ciej litery:
dodatkowa: pusta
a : OLA ALA ELA
b : KUBA
n : JAN
r : MAREK
3 scalenie:
OLA ALA ELA KUBA JAN MAREK
Zauważmy, że w tym kroku połączony ciąg jest posortowany zgodnie z alfabetyczną kolejnością trzecich liter wyrazów.
4 rozrzut – według 2-giej litery:
a : JAN MAREK
l : OLA ALA ELA
u : KUBA
4 scalenie:
JAN MAREK OLA ALA ELA KUBA
5 rozrzut – według 1-szej litery:
a : ALA
e : ELA
j : JAN
k : KUBA
m : MAREK
o : OLA
5 scalenie: (i ostatnie)
ALA ELA JAN KUBA MAREK OLA

Istotę prawidłowego działania algorytmu sortowania leksykograficznego stanowi wykorzystanie specyficznej struktury danych, jaką jest kolejka. Wstawianie elementów tylko na koniec kolejki a pobieranie ich tylko z początku pozwala zachować uporządkowanie (począwszy od pewnego miejsca) osiągnięte we wcześniejszych krokach algorytmu.

Koszt tego algorytmu jest rzędu: liczba liter alfabetu + suma długości wszystkich sortowanych słów. Jest to doprawdy niewielki koszt w porównaniu z liczbą operacji, jakiej trzeba by dokonać, aby posortować słowa przy użyciu jednego z algorytmów posługujących się tylko porównywaniem obiektów. Dla przykładu program w polskim Logo sortujący leksykograficznie 40 słów na mikrokomputerze ZX Spectrum działał 1.5 minuty. Natomiast program (także w polskim Logo i także na mikrokomputerze ZX Spectrum) sortujący 40 tych samych słów w ten sposób, że brał kolejne słowo i wstawiał je w odpowiednie miejsce tworzonego uporządkowanego ciągu, działał 6 minut.

Wynik ten nie wymaga chyba komentarza, poza jednym może – nawet ten lepszy czas nie był wcale rewelacyjny. Nie jest to jednak wina algorytmu, lecz tego, że został on zapisany w Logo, który to język jest doskonałym narzędziem dydaktycznym, ale zupełnie nie nadaje się do praktycznego zastosowania przy naprawdę szybkim sortowaniu. Na usprawiedliwienie wypada jeszcze dodać, że programy były testowane przy mocno zajętej pamięci i w trakcie ich wykonania często następowało "odśmiecanie" pamięci. Możliwość "odśmiecania" została wprowadzona do interpretera polskiej wersji Logo. Oryginalny angielski interpreter Logo na ZX Spectrum przy przepełnieniu pamięci po prostu przestawał reagować na jakiekolwiek rozkazy.

Pora przedstawić program sortujący leksykograficznie. Ma on następującą postać:

```
OTO SłOWNIK :słowa
JEŚLI ( DłUGOść :słowa ) < 2 [wy
nik :słowa]
PRZYPISZ "maxdł 0
WYNIK SłOWA.UP PORZąDEK PARY :sł
owa :maxdł
JUż
```

Lista o nazwie "słowa" zawiera wejściowy ciąg słów do posortowania. Procedura "słownik" zwraca uporządkowany ciąg słów.

```
OTO SłOWA.UP :lista.par
JEŚLI PUSTE? :lista.par [wynik [
]]
WYNIK ZDANIE OST PIERW :lista.pa
r SłOWA.UP BEZPIERW :lista.par
JUż
```

▶ 36

35 ◄

Procedura "słowa. up" tworzy ciąg słów z ciągu par: długość słowa, słowo.

```
OTO PORZĄDEK :lista.par :nn
JEŚLI :nn = 0 [wynik scalone.kol
ejki 0]
INICJALIZACJA.KOLEJEK
DYSTRYBUCJA :lista.par :nn
WYNIK PORZĄDEK SCALONE.KOLEJKI 0
 :nn - 1
JUŻ
```

Procedura "porządek" zwraca ciąg par jw. posortowany zgodnie z alfabetycznym porządkiem słów.

```
OTO PARY :lista
JEŚLI PUSTE? :lista [wynik []]
JEŚLI ( DŁUGOŚĆ PIERW :lista ) >
 :maxdł [przypisz "maxdł długość
 pierw :lista]
WYNIK NAP ZDANIE DŁUGOŚĆ PIERW :
lista PIERW :lista PARY BEZPIERW
 :lista
JUŻ
```

Procedura „pary" działa odwrotnie do "słowa. up". Dla każdego słowa z wejściowej listy określa jego długość i zwraca listę par w postaci: [długość słowa słowo]

```
OTO INICJALIZACJA.KOLEJEK
INIT 97
PRZYPISZ "dodatkowa []
JUŻ

OTO INIT :kod
JEŚLI :kod = 123 [stop]
PRZYP ZNAK :kod []
INIT :kod + 1
JUŻ
```

Procedury: inicjalizacja.kolejek i init tworzą puste kolejki. Nazwami kolejek są litery określone przy pomocy kodu ASCII (97 to kod "a", 123 – kod "z"). Jak łatwo zauważyć, stosowane słowa powinny składać się tylko z małych liter. Jeśli chcemy porządkować słowa złożone z innych znaków, wystarczy zmodyfikować w odpowiedni sposób procedurę init.

```
OTO DYSTRYBUCJA :lista.par :nn
JEŚLI PUSTE? :lista.par [stop]
JEŚLI ( PIERW PIERW :lista.par )
 < :nn [przypisz "dodatkowa nak
pierw :lista.par :dodatkowa] [pr
zypisz elem :nn ost pierw :lista
.par nak pierw :lista.par wartoś
ć elem :nn ost pierw :lista.par]

DYSTRYBUCJA BEZPIERW :lista.par
:nn
JUŻ
```

Procedura "dystrybucja" rozrzuca słowa do kolejek według litery n-tej w słowie podawanej przez "elem :nn słowo".

```
OTO ELEM :n :s
JEŚLI LUB :n < 1 :n > DŁUGOŚĆ :s
 [wynik "]
JEŚLI :n = 1 [wynik pierw :s]
WYNIK ELEM :n - 1 BEZPIERW :s
JUŻ

OTO SCALONE.KOLEJKI :nn
WYNIK ZDANIE :dodatkowa SCAL.KOL
 :nn
JUŻ

OTO SCAL.KOL :nn
JEŚLI :nn = 26 [wynik []]
WYNIK ZDANIE WARTOŚĆ ZNAK :nn +
97 SCAL.KOL :nn + 1
JUŻ
```

Procedura "scalone.kolejki" zwraca ciąg utworzony przez scalenie wszystkich kolejek począwszy od dodatkowej przez "a", "b" aż do "z".

AGNIESZKA KASPRZYCKA

(w jednym z następnych numerów cz. II o sortowaniu wewnętrznym w ogólnym przypadku – przy użyciu jedynie porównań obiektów).

Kolejka jest specjalnym rodzajem listy. Elementy są wstawiane do kolejki z jednego końca, a usuwane z drugiego. W języku angielskim (informatycznym) często zamiast queue (kolejka) używa się określenia FIFO od słów first in first out, co oznacza w wolnym przekładzie: pierwszy wchodzi, pierwszy wychodzi. Nasuwa się analogia z kolejką na przykład w sklepie – ten, kto pierwszy przyszedł, jest obsłużony jako pierwszy.

PROGRAMOWANIE GIER LOGICZNYCH

Alpha-beta pruning

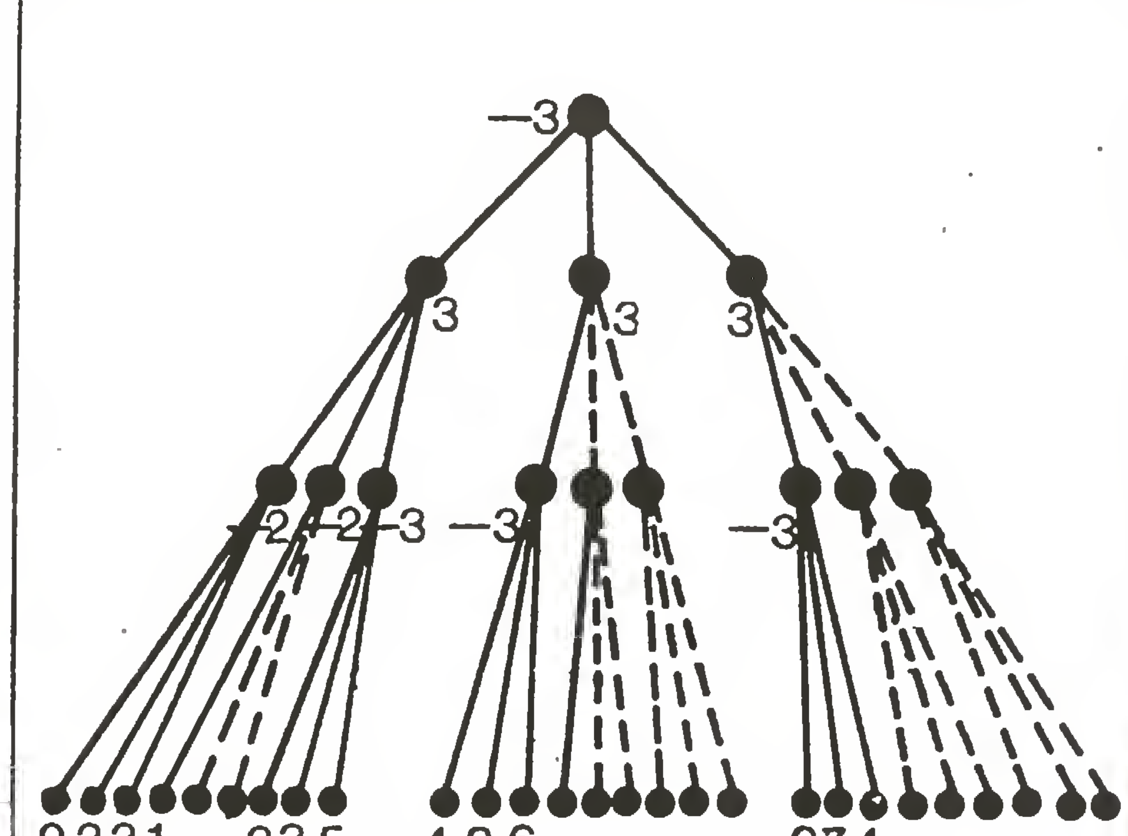
```
main()
{int i, j, val;
for(i=P3+1;i<P10;i++) tab[P12][i]=P3+P6;
for(i=0;i<3;i++){pi[i][0]=-(i+1);pi[i][1]=0; pi[i][2]=i+1;
                 pj[i][0]=pj[i][1]=pj[i][2]=1+i;}
for(i=P3+1;i<=P3+P6;i++)
    for(j=P3+1;j<P10;j++) MASKA[i][j]=1;
rep : printf("Podaj wysokość drzewa \n");
      scanf("%d",&poziom);
      if(!(poziom>=1 && poziom<20)) goto rep;
poziom++; kol=1;
printf("Czy chcesz zagrać pierwszy? (1 lub 0) \n");
scanf("%d", &i);
if(i) {j=czytajruch(); WSTAW(j+P3,kol); kol=-kol;}
WSTAW(4+P3,kol);kol=-kol;drukuj(4+P3);
loop: j= czytajruch(); j+=P3;
      WSTAW(j,kol);  drukuj(j); win=ocenapoz(j);
      if(win){printf("PRZEGRAŁEM \n"); exit();}
      kol=-kol;
      val=alphabeta();
      if(val == 0) {printf("PODDAJĘ SIĘ \n"); exit();}
      WSTAW(val,kol);  drukuj(val);
      win=ocenapoz(val);
      if(win){printf("WYGRAŁEM \n"); exit();}
      kol=-kol; goto loop;
 }
 czytajruch()
 {int j;
 for(j=P3+1;j<P10;j++) if(tab[P12][j]> P3) goto podaj; remis();
podaj:  printf("Podaj ruch (1 do 7) \n");
 scanf("%d", &j);
 if(j<1 || j>7 || tab[P12][j+P3]<=P3) goto podaj; return(j);}
 drukuj(j)
 int j;
 {int p,q;
  for(p=P3+1;p<=P3+P6;p++)
    { for(q=P3+1;q<P10;q++)
      if(tab[p][q]==0) printf("| |");
      else (tab[p][q]==1? printf("|X|") :printf("|O|"));
      printf("\n");}
 printf("\n Ostatni ruch w kolumnie %d \n",j-P2);}
remis()
{printf("REMIS  \n");exit();}
focena(i,j)
int i,j;
{int p, k, s, ti, tj;
int v[3];
s=(tab[i+1][j]==-kol)? 2:0;
for(p=0;p<3;p++)
    {v[p]=0; for(k=0;k<3;k++)
                {ti=i+pi[k][p]; tj=j+pj[k][p];
                 if(MASKA[ti][tj])
                    {if(tab[ti][tj]==kol) break;
                     else v[p]+=(tab[ti][tj]? P5 : P2);}
                 else break;}}
for(p=0;p<3;p++)
    { for(k=0;k<3;k++)
                {ti=i-pi[k][p]; tj=j-pj[k][p];
                 if(MASKA[ti][tj])
                    {if(tab[ti][tj]==kol) break;
                     else v[p]+=(tab[ti][tj]? P5 : P2);}
                 else break;}}
for(p=0;p<3;p++) s+=v[p];
return(s);}
```

```
#include <stdio.h>
#define P3 2
#define P12 12
#define P10 10
#define P6 6
#define P5 5
#define P2 2
#define P100 100
int kol, win, first, poziom;
int tab[13][13], MASKA[13][13];
int pi[3][3],pj[3][3];
#define WSTAW(k,c)  (tab[tab[P12][k]][k]=c, tab[P12][k]--)
#define USUN(k)     (tab[P12][k]++, tab[tab[P12][k]][k]=0)
next(kl)
int kl;
{int k;
for(k=kl+1;k<P10;k++)
    if(tab[P12][k]>P3) return(k);
return(NULL);}
generuj(l)
int l;
 { first=(l>poziom || win)? NULL : next(P3);}
ocenapoz(j)
int j;
{int k, i;
i=tab[P12][j]+1;
if(tab[i+3][j]==kol && tab[i+2][j]==kol && tab[i+1][j]==kol)
                            return(P100);
for(k=0;k<3;k++)
 {  if(tab[i+pi[0][k]][j+pj[0][k]] != kol) goto E3;
    if(tab[i+pi[1][k]][j+pj[1][k]] != kol) goto E2;
    if(tab[i+pi[2][k]][j+pj[2][k]] != kol) goto E1;
    return(P100);
E3: if(tab[i-pi[2][k]][j-pj[2][k]] != kol) continue;
E2: if(tab[i-pi[1][k]][j-pj[1][k]] != kol) continue;
E1: if(tab[i-pi[0][k]][j-pj[0][k]] != kol) continue;
    return(P100);}
return(0);}
alphabeta()
{int ply,a[20],r[20], mem=0;
ply=2; a[0]=a[1]=-P100;
first=next(P3); if(first==NULL) remis(); else goto ini;
nowe_ruchy:  generuj(ply);
ini:         r[ply+1]=first;
if(r[ply+1]==NULL){
a[ply]=ply-(win?win:focena(tab[P12][r[ply]]+1,r[ply]));}
   else
   {    a[ply]=a[ply-2];
nizej:  ply++; WSTAW(r[ply],kol); win=ocenapoz(r[ply]); kol=-kol;
         goto nowe_ruchy;
zbadaj: if(-a[ply+1]>a[ply])
          {a[ply]=-a[ply+1];
           if(a[ply+1]<=a[ply-1]) {USUN(r[ply+1]);
                                   goto koniec_poziomu;}}
        USUN(r[ply+1]);
        r[ply+1]= next(r[ply+1]);
        if(r[ply+1]) goto nizej; }
koniec_poziomu:  ply--; if(ply>=2){ kol=-kol;
                 if(ply==2 && -a[3]>a[2]) mem=r[3];
                 goto zbadaj;}
return(mem);}
```

W poprzednim odcinku mówiliśmy o metodzie generowania ruchu na podstawie drzewa gry, zwanej minimaks. Poważną niedoskonałością tej metody jest sprawdzanie zawsze całego drzewa. Bywają sytuacje, kiedy bez wpływu na ostateczny wynik, można szybciej zakończyć przeszukiwanie pewnych fragmentów drzewa. Na przykład na rys. 2 z poprzedniego odcinka można było nie badać ostatnich dwóch liści (węzłów końcowych) o wartościach 3 i 5. Dlaczego? Mam nadzieję wyjaśnić to w dalszym ciągu artykułu.

Obecnie najlepszą metodą "obcinania" zbędnych gałęzi jest algorytm zwany alpha-beta pruning. Pruning oznacza obcinanie, a alpha i beta (dalej będę pisał alfa zamiast alpha) to dwie liczby (zmienne), przy pomocy których dokonuje się obcięcia. Spójrzmy na rys. 1 prezentujący wynik działania tego algorytmu. Linie przerywane wskazują na węzły, które zostały pominięte (obcięte). Łatwo zauważyć, jak duże oszczędności czasowe może dać ta metoda.

Powiedzmy w końcu na czym ona polega. Istotne jest, żeby pamiętać zasadę działania algorytmu minimaks, która omówiona została w poprzednim odcinku.



Spójrzmy na pierwsze trzy liście drzewa: 9, 2 i 3. Pamiętamy, że każdemu węzłowi odpowiada pewna pozycja w grze. Zalety pozycji wyrażają liczby przypisane liściom przez funkcję oceniającą. Ustalmy jeszcze rzecz następującą. Z każdego węzła odchodzą trzy gałęzie, które wyrażają trzy i tylko trzy możliwe ruchy gracza. Nazwijmy je odpowiednio L, S, P od gałęzi lewej, środkowej i prawej. W ten sposób każdy węzeł daje się opisać jako ciąg liter L, S i P, które odpowiadają ruchom, jakie trzeba wykonać, aby do danej pozycji (węzła) dojść. Na przykład węzeł LPP, to liść o wartości 5.

Korzeń reprezentuje konkretny ruch (L, S lub P), jaki ma zostać oceniony za pomocą tego drzewa. Stąd poziomy B i D odpowiadają pozycjom po ruchu przeciwnika. Funkcja oceniająca rozpatruje pozycje z "naszego" punktu widzenia, dlatego też pozycja o minimalnej wartości jest dla naszego przeciwnika najkorzystniejsza. Dlatego z liczb 9, 2 i 3 zapisujemy w poziomie C wartość 2, ale ze zmienionym znakiem. Czyni to z metody minimaks tzw. megamaks i znacznie ułatwia zapis algorytmu: zamiast na przemian szukać liczby najmniejszej i największej, zawsze szukamy najmniejszej, po czym zapisujemy ją wyżej ze zmienionym znakiem.

Węzeł LL otrzymuje wartość -2. Zostaje ona także zapamiętana pod zmienną beta. Następnie badana jest pozycja LSL, która oceniona została na 1 punkt. Dlaczego możemy pominąć dalsze węzły tego poddrzewa – LSS i LSP?

Proszę pamiętać, że zapisujemy wyżej wartość najmniejszą, która dla tego poddrzewa będzie mniejsza bądź równa 1. A skoro tak, to po zmianie znaku w poziomie C wartość ta i tak przegra w konkurencji zapisania do poziomu B, gdyż w węźle LL mamy już -2. Ujmując tę zależność w postaci warunku występującego w algorytmie (patrz procedura alfabeta dołączonego programu, drugi wiersz pod etykietą zbadaj) powiemy, że najmniejsza dotychczas wartość z tego poddrzewa – w tym przypadku 1 – jest mniejsza lub równa minus beta. W węźle LS zapisujemy wartość beta i ignorujemy resztę drzewa.

Zależności pomiędzy wartościami węzłów różnych poziomów badane są w algorytmie przy pomocy zmiennych alfa i beta. Stanowią one odpowiednio dolne i górne ograniczenie akceptowanych wartości węzłów. Stąd zwykłe porównania decydują o tym, które gałęzie zostaną "obcięte" a które nie. Alfa i beta zmieniają się w zależności od poziomu i poddrzewa.

Inaczej jest z poddrzewem LP, czyli węzłami LPL, LPS i LPP. Trzeba je wszystkie zbadać, gdyż najmniejsza wartość tego poddrzewa ma szansę "zajść" wysoko – nie zachodzi warunek przytoczony wyżej, który spowodował "obcięcie" gałęzi LSS i LSP. Po sprawdzeniu całego poddrzewa LP, pojawia się nowy "lider" w postaci wartości 3, który zdobywa kolejne poziomy aż do samego korzenia. Zmienna alfa przyjmuje teraz wartość – 3, co oznacza, że żadna mniejsza liczba nie ma prawa znaleźć się w korzeniu.

W ten sposób zakończone zostało przeszukiwanie lewego poddrzewa. Przyjrzyjmy się teraz węzłom LPL, LPS, LPP o wartościach 4, 8 i 6. Widzimy, że po znalezieniu najmniejszej z nich (4), całe środkowe poddrzewo zostało pominięte. Czy słusznie? Wartość 4 z poziomu D zapisana byłaby do poziomu C (ze zmienionym znakiem oczywiście) i tam miałaby duże szanse na "awans" jeszcze wyżej. Lecz na poziomie B wartość 3 zapamiętana w węźle L nie daje jej żadnych szans na zdobycie "szczytu" w postaci poziomu A. "Niech zostanie na miejscu, skoro nie może dojść do samego końca" – wydaje się być tajemnicą sukcesu tej metody. Analogicznie jest z poddrzewem prawym.

Jak w kategoriach ruchów poszczególnych graczy wytłumaczyć redukcję środkowego poddrzewa? Na bazie założenia, że przeciwnik ma trochę rozsądku. Bo jeśli będzie na tyle naiwny, że w pozycji reprezentowanej przez korzeń zagra ruch S (z poziomu A), wówczas my grając L (z poziomu B), pozostawimy mu co najwyżej ruch L (z poziomu C) dający pozycję o wartości 4. Jest to ewidentna strata dla niego, bo gdyby zagrał L na samym początku, to najprawdopodobniej osiągnąłby pozycję o wartości 3 (czyli lepszą).

Algorytm ten skonstruowali Donald E. Knuth i Ronald W. Moore w 1975 roku[1]). Od tego czasu jest zupełnie nie do pomyślenia stosowanie metody minimaks bez optymalizacji, jaką dostarcza alfa-beta pruning.

Od czego zależy ilość "obciętych" gałęzi? Jest to bardzo istotna sprawa. W przypadku tego rysunku zbadanych zostało około połowy wszystkich węzłów, gdy na rys. 2 z poprzedniego odcinka można było pominąć tylko dwa węzły. Różnice te powoduje uporządkowanie węzłów. Najkorzystniejszym układem jest taki, w którym najlepszy ruch jest rozpatrywany na początku. Zauważmy, że na rys. 1 w korzeniu drzewa znalazła się wartość pochodząca z lewego poddrzewa. Był to więc układ bliski optymalnemu. W programach grających w szachy stosuje się krótkie wstępne przeszukiwanie, którego celem jest tylko uporządkowanie węzłów. Dopiero potem uruchamia się właściwy algorytm alfa-beta, który w pierwszej kolejności bada ruch "prawdopodobnie" najlepszy. W sumie alfa-beta pruning umożliwia przeszukiwanie większych drzew w tym samym czasie, co istotnie wpływa na siłę i skuteczność gry komputerów.

```
function alphabeta: integer;
label 1111, 2222, 3333, 4444, 5555;
{etykieta 1111 to nowe_ruchy.
          2222 to ini,
          3333 to nizej,
          4444 to zbadaj,
          5555 to koniec_poziomu.}
var
   ply, mem: integer;
   a:     array[0..20] of integer;
   r:     array[0..20] of integer;
begin
mem=0; ply=2;
a[0]=a[1]=-P100;
first=next(P3); if first=NULL then remis else goto 2222;
1111:  generuj(ply);
2222:  r[ply+1]=first;
if r[ply+1]=NULL then
begin
     if win=0 then a[ply]=ply-focena(tab[P12,r[ply]]+1,r[ply])
     else a[ply]=ply-win;
end
else
begin
    a[ply]=a[ply-2];
3333:     ply=ply+1; WSTAW(r[ply],kol); win=ocenapoz(r[ply]);
          kol=-kol;
          goto 1111;
4444:     if -a[ply+1]>a[ply] then
    begin
       a[ply]=-a[ply+1];
       if a[ply+1]<=a[ply-1] then
       begin
         USUN(r[ply+1];
         goto 5555;
       end;
    end;
    USUN(r[ply+1]);
    r[ply+1]= next(r[ply+1]);
    if r[ply+1] then goto 3333;
end;
5555:     if ply>=2 then
          begin
               kol=-kol;
               if (ply=2) and (-a[3]>a[2]) then mem=r[3];
               goto 4444;
          end;
alfabeta = mem;
end.
```

Teraz zajmiemy się przykładem programu. Napisany on został w języku C na mikrokomputery zgodne z IMB PC. Kompilatory C są dostępne na wszystkich bardziej popularnych mikro. Programy w tym języku mogą być łatwo przenoszone z jednego komputera na inny, gdyż różnice między poszczególnymi wersjami są minimalne. Na przykład na Spectrum 48 K nie dopuszcza się parametrów w "makrach", tak więc "USUN" i "WSTAW" trzeba byłoby zamienić na procedury.

Program ten realizuje algorytm alfa-beta pruning dla gry kółko i krzyżyk z ciążeniem. Oto krótki opis. Gra się na planszy 6 na 7; wygrywa zawodnik, który ustawi cztery kółka bądź krzyżyki w pionie, poziomie lub na ukos. Specyfiką tej gry jest to, że można wykonać ruch w dowolnej z siedmiu kolumn, ale w rzędzie najniższym z możliwych. Upraszcza to znacznie generowanie legalnych ruchów, gdyż jest ich co najwyżej siedem (każda kolumna). W tablicy o nazwie "a" przechowywane są wartości alfa, minus beta, najmniejszy z danego poddrzewa o przeciwnym znaku oraz aktualna wartość rozpatrywanego liścia, w elementach odpowiednio: "a[ply-2]", "a[ply-1]", "a[ply]" o "a[ply+1]" dla każdego poziomu "ply". W "r[ply+1]" zapamiętany jest numer kolumny, w której wykonany został ostatni ruch – ciąg liczb tablicy "r" odpowiada zapisowi typu LSP.

Mamy dwie funkcje: "ocenapoz" oraz "focena", analizujące pozycje. Pierwsza sprawdza czy nie nastąpiła wygrana któregoś z graczy, druga pełni rolę funkcji oceniającej. Jak już mówiliśmy, dobra funkcja oceniająca jest podstawą dobrej gry komputera. Jednakże do wymyślenia takiej funkcji potrzeba eksperta, który umie ocenić pozycję i wie, co liczy się in plus a co in minus. Ja do takich nie należę. Jednak mimo to program gra szybko (na IBM PC) i nieźle na poziomach 4 i 5. Zachęcam Czytelników do zastanowienia się i napisania własnej funkcji oceniającej, co jest zajęciem twórczym i dającym satysfakcję. Na szybkim komputerze można w ten sposób stworzyć silnie grający program.

Poważnym mankamentem jest grafika i dlatego chciałbym przeprosić tych, którzy nastawiają się raczej na przepisywanie i użytkowanie programu niż na dopracowanie i wzbogacenie.

Najważniejszy w tym przykładzie jest algorytm alfa-beta pruning. Realizuje go procedura o nazwie "alfabeta". Zmienna "mem" zapamiętuje numer kolumny, która spowodowała ostatnią zmianę w korzeniu drzewa. Numer ten jest wynikiem działania procedury i zarazem wskazuje następny ruch.

Chętnie dowiedziałbym się o skuteczności i działaniu nowych funkcji oceniających, jeżeli takie stworzycie i zechcecie udostępnić innym. Dotyczy to również całego programu.

JANUSZ KRASZEK

1) "An Analysis of Alpha-Beta Pruning", Donald E. Knuth, Ronald W. Moore, Volume 6, Number 4, 1975 ARTIFICIAL INTELLIGENCE.

LISTY

Jeszcze nie ukazał się numer drugi "Komputera" (tak, cykl wydawniczy trwa niestety 6 tygodni i te słowa musiały być napisane na początku maja), a czytelnicy dokonują już szerszej oceny naszej "produkcji".

Z uwagi na obszerność niektórych listów musimy, niestety, zastrzec sobie prawo skracania tych, które będziemy zamieszczać na naszych łamach.

• • •

Szanowny Panie Redaktorze

Piszę w pewnym pośpiechu, mając możliwość tylko krótkiego przedyskutowania z przyjaciółmi niektórych z poruszonych niżej zagadnień. Bardzo proszę więc Pana o wyrozumiałość, jeżeli znajdzie Pan sformułowania niezbyt precyzyjne. Oto tezy:

– Wbrew zapowiedziom prenumerata nie jest przyjmowana na II półrocze i nikt w "Ruchu" nie wie, kiedy będzie;

– Nie rozumiem Szanownej Redakcji: dlaczego mówicie o kasetach z programami? Kasety są b. potrzebne, ale od tego powinno być wyspecjalizowane przedsiębiorstwo – elastyczne, rzutkie, dostosowane do specyfiki branży;

– W kręgach użytkowników informatyki mówi się o przetłumaczeniu, poprawieniu i wydaniu dokumentacji do typowego oprogramowania. Oprogramowanie warte jest tyle ... co jego dokumentacja!

– Biblioteka ABC "Komputera" jest dość dobrym pomysłem. Uważam, że powinna być regularna i podlegać prenumeracie. Kwestią sporną pozostaje poziom tej serii. Uważam, że powinny być to zeszyty zastępujące podręczniki dla szkół, klubów – ale nie ABC!

– Czytając skład redakcji miło widzieć nazwiska osób spoza Stolicy. Dobrze byłoby podać, z jakich są miast i jak nawiązać z nimi kontakt;

– Pilnym i ważnym tematem jest metodologia programowania. Poznanie zasad racjonalnego, wydajnego tworzenia poprawnego oprogramowania jest zadaniem o wielkiej wadze społecznej. Materiałów na ten temat w języku polskim jest mało. Ludzie piszą programy użytkowe i powinni wiedzieć, JAK to powinno się robić;

– Koniecznie podawajcie daty, miejsca i tematy kursów, szkoleń, konferencji, wystaw oraz innych spotkań z branży;

– Nie publikujcie recenzji książek itp. Publikujcie ZAMIARY wydawców. To ułatwi zaopatrzenie w nową literaturę;

– Sądzę, że warto informować Czytelników o wybranych nowościach wydawniczych firm zagranicznych – ze Wschodu, Zachodu i Południa;

– Konieczne jest poświęcenie nieco miejsca na krótki przegląd najważniejszych artykułów, które pojawiły się w czasopismach innych krajów – jako minimum: tytuł BYTE powinien być w kilku zdaniach omawiany na bieżąco;

– Pilną sprawą jest rozpoczęcie i konsekwentne kontynuowanie bibliografii "Mikrokomputer w szkole";

– Warto pamiętać o prawnych aspektach kopiowania i tworzenia oprogramowania. To ważny element kultury informatycznej i etyki zawodowej;

– Najpilniejszą sprawą jest materiał o pamięciach na dyskach elastycznych. Ich budowa, właściwości, eksploatacja, konserwacja i wskazówki do samodzielnych prostych napraw. Ileż od tego zależy!

Łączę wyrazy szacunku dla Pana i Współpracowników

Jacek Żebrowski
Łódź

Powyższy list jest jedną z pierwszych recenzji, jakie otrzymaliśmy. Porusza sprawy raczej typowe, sugerowane nam także w innych omówieniach.

Jak zawsze w przypadku powstania nowego czasopisma są kłopoty z prenumeratą. Według zapewnień kompetentnych urzędów sprawa jest już sfinalizowana, tak że nie powinno być z tym trudności (oczywiście do momentu wyczerpania limitu).

Do tej pory nic nam nie wiadomo, aby miało powstać przedsiębiorstwo produkcji i dystrybucji programów mikrokomputerowych na kasetach magnetofonowych. Na razie więc my planujemy zapełnienie tej luki, postaramy się przy tym być rzutcy i elastyczni.

Z informacji nam dostępnych wynika, że podobne prawa dotyczą kopiowania programów jak i powielania ich dokumentacji. Rzeczywiście nawet najlepszy program bez instrukcji jest niewiele wart (vide casus COPY-COPY, ale tu sprawa była prostsza, bo autora mieliśmy "pod ręką").

W serii ABC "Komputera" zamierzamy wydać między innymi opisy sprzętu i oprogramowania najpopularniejszych w naszym kraju mikrokomputerów osobistych. Dołożyłymy wszelkich starań (a właściwie autorzy), aby nie było to tylko podstawowe ABC. Czy seria będzie dostępna w prenumeracie, trudno jeszcze w tej chwili powiedzieć.

W obecnej chwili skład redakcji dopiero się stabili-

▶ 40

39 ♦

zuje. Wszystkie zmiany będą na bieżąco uwzględniane w stopce redakcyjnej. Postaramy się zamieścić także bliższe informacje o naszych współpracownikach.

Redakcja docenia w pełni rolę poprawności metodologii programowania. Na marginesie szerzenia kultury informatycznej, będącego jednym z założeń naszej działalności, zajmiemy się także propagowaniem poprawnego stylu programowania.

Będziemy starali się podawać informacje o wszelkiego rodzaju kursach czy wystawach, ale musimy je otrzymywać odpowiednio wcześniej, aby w momencie ukazywania się w druku były aktualne (cykl wydawniczy!).

Co do zamiarów wydawnictw, to sprawa jest o tyle skomplikowana, że częstokroć wydawnictwa nie mogą spełnić swoich zamiarów z oczywistych powodów poligraficznych.

Przegląd nowości z różnych stron świata jest stałym punktem naszego "menu".

Wspominaliśmy już w tym miejscu o upowszechnianiu edukacji mikrokomputerowej, gwoli ścisłości dodajmy tylko, że jest to następny z naszych podstawowych celów.

Zamierzamy w niedługim czasie podjąć kwestie ochrony praw autorskich i co za tym idzie zagadnień związanych z rozpowszechnianiem programów.

Materiały dotyczące niektórych urządzeń peryferyjnych (floppy dyski) już są, czekają tylko na swoją kolejkę.

REDAKCJA

• • •

W pierwszym numerze "Komputera" wydrukowany został krótki program w języku LOGO pt. "Kometa Halleya". Z przyczyn częściowo od nas niezależnych w tekst programu wkradły się błędy, uniemożliwiające poprawne jego uruchomienie. W związku z tym podajemy pełny i poprawny jego listing. Czytelników i autora programu bardzo przepraszamy.

```
TO KOMETA
HT CS PD
SETBG Ø SETBR Ø SETTC [Ø 6]
INTRODUKCJA
CS
SETCUR [1Ø 3]
PR [PER COMPUTER]
SETPC 7 KROPKI.
SETCUR [12 19]
PR [AD ASTRA]
SETPC 6 KROPKI..
SETTC [Ø 4] SETCUR [14 21]
PR [CZPL]
MAKE "x - 9Ø + RANDOM 45
MAKE "y - 3Ø + RANDOM 15
SETPC 4 KROPKI...
END

TO INTRODUKCJA
SETCUR [1Ø 2] PR [KOMETA©ZPL]
SETCUR [3 7]
PR [Program startuje po]
PR [nacisnieciu dowolnego klawis
za.]
PR [] PR []
PR [UWAGA: W trzeciej minucie]
PR [dzialania programu]
PR [widoczna kometa Halleya!]
WAIT 12Ø
MAKE "start RC
END

TO KROPKI.
REPEAT 5Ø + RANDOM 1ØØ [dot se (
 random 12Ø ) - 6Ø ( random 5Ø )
 - 25]
END

TO KROPKI..
REPEAT 1ØØ + RANDOM 2ØØ [dot se
( random 18Ø ) - 9Ø ( random 76
) - 38]
END

TO KROPKI...
REPEAT 1Ø + RANDOM 1Ø [dot se (
random 24Ø ) - 12Ø ( random 1ØØ
) - 5Ø]
SETPC 7
DOT SE :x + RANDOM 7 :y + RANDOM
 4
SETPC 4 + RANDOM 2
MAKE "x :x + RANDOM 1 + RANDOM 2

MAKE "y :y + RANDOM 2
IF :y > 47 [toplevel]
KROPKI...
END
```


KONKURS IM. JERZEGO TRYBULSKIEGO

W kryzysowych latach 1981-1982 znacznie zmniejszyła się liczba wdrożonych prac dotyczących zastosowań informatyki w gospodarce narodowej, zmniejszył się wymiar prac projektowo-programowych, nastąpiła stagnacja ilościowa i jakościowa w odniesieniu do wyposażenia sprzętowego ośrodków informatyki. Nastąpił jednym słowem znaczny regres w rozwoju polskiej informatyki.

Jednym ze sposobów wprowadzania, rozwoju i popularyzacji nowych metod i środków informatyki w różnych dziedzinach zastosowań może stać się ogólnopolski konkurs na najlepsze wdrożone prace z dziedziny zastosowań informatyki w gospodarce narodowej, organizowany przez Polskie Towarzystwo Informatyczne.

Podstawowym celem konkursu jest stymulacja szybkiego rozwoju zastosowań informatyki w różnych dziedzinach gospodarki narodowej. I Walny Zjazd PTI, chcąc uczcić pamięć Jerzego Trybulskiego, jednego z założycieli Towarzystwa, oraz doceniając Jego wielki wkład w rozwój zastosowań informatyki postanowił, aby organizowany konkurs nazwać Jego imieniem.

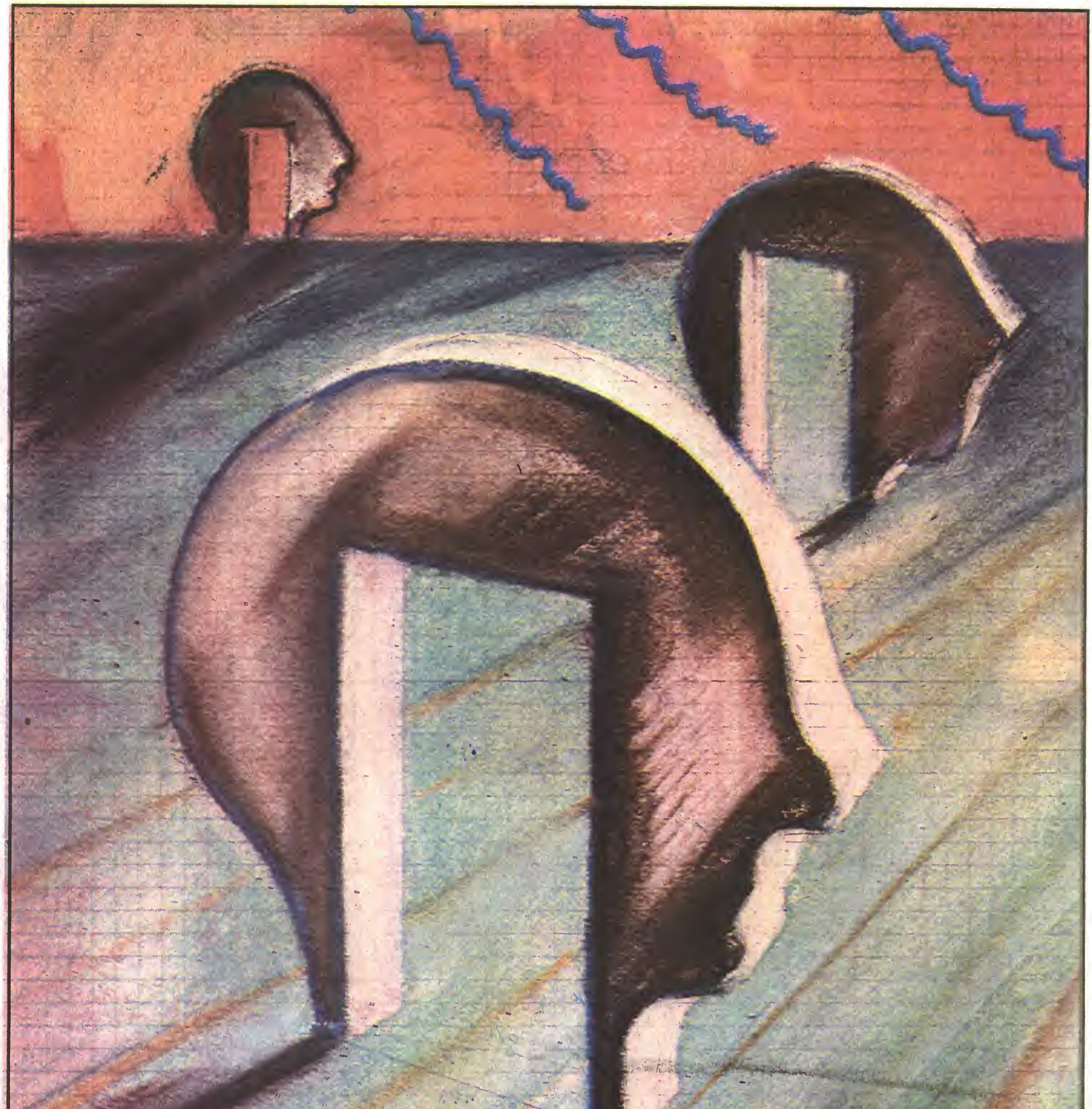
Opracowanie projektu regulaminu konkursu i zorganizowanie go Zarząd Główny Polskiego Towarzystwa Informatycznego powierzył Dolnośląskiemu Oddziałowi PTI we Wrocławiu.

WARUNKI KONKURSU

- Prace zgłoszone na konkurs mogą dotyczyć tylko rozwiązań informatycznych wdrożonych w jednostkach gospodarki uspołecznionej i nieuspołecznionej w kraju i za granicą.
- W konkursie mogą wziąć udział obywatele Polski, który są autorami prac wykonanych na terenie kraju.
- Komisja konkursowa będzie oceniać tylko rozwiązania informatyczne bieżąco eksploatowane przez użytkownika co najmniej jeden rok.
- Zgłoszone na konkurs wdrożone prace informatyczne będą oceniane według kryterium uzyskanych wymiernych i niewymiernych efektów ekonomicznych.
- Przedmiotem oceny nie będą wdrożenia dotyczące:
  - języków oprogramowania;
  - systemów operacyjnych;
  - szeroko rozumianej technologii komputerowego przetwarzania, oprogramowania narzędziowego i pomocniczego.
- Prawo zgłaszania na konkurs prac spełniających powyższe warunki mają dyrektorzy, kierownicy jednostek autorskich lub eksploatujących oraz indywidualni autorzy rozwiązań informatycznych.
- Termin nadsyłania kart zgłoszeń na konkurs ustala się na dzień 30 września roku kalendarzowego poprzedzającego rok Walnego Zjazdu PTI*.
- Zgłoszenia należy nadsyłać pod adresem: Polskie Towarzystwo Informatyczne, Oddział Dolnośląski Plac Grunwaldzki 9, 50-370 Wrocław (z dopiskiem "Konkurs im. Jerzego Trybulskiego").

(g)

* Walny Zjazd PTI odbędzie się w 1987 roku.



ZWIELOKROTNIONA PAMIĘĆ EKRANU W ZX SPECTRUM

Ekran mikrokomputera ZX Spectrum mieści zaledwie 22-24 linie po 32 znaki w każdej. Jest to niewiele – w praktyce potrzebny jest szybki dostęp do większej ilości wyświetlanej na ekranie informacji. Jedynym wyjściem jest podział tej informacji między kilka ekranów. Przykładem mogą być programy edukacyjne lub użytkowe, przeznaczone dla niewprawnego użytkownika. W razie wątpliwości lub trudności w obsłudze programu użytkownik musi mieć możliwość sprawdzenia na ekranie pojedynczym wciśnięciem klawisza odpowiedniego tekstu objaśniającego lub przykładów poprawnej konwersacji z maszyną. Po odczytaniu wyjaśnień kolejne wciśnięcie klawisza powinno przywracać poprzedni wygląd ekranu, pozwalając na kontynuację pracy. Podobna sytuacja występuje w grach komputerowych, gdy trzeba zobrazować na zmianę mapę kosmicznego sektora, konsolę pilota gwiazdolotu i tekst objaśnień.

Jeśli mamy do czynienia wyłącznie z tekstami, możemy rozwiązać problem w języku BASIC, budując za każdym razem od nowa cały ekran. Postępowanie takie jest męczące dla oczu, wiąże się z opóźnieniami, zaś w przypadku użycia grafiki najczęściej niemożliwe. Znacznie lepiej posłużyć się prościutkim programem maszynowym, umożliwiającym "przełączanie" między dwoma lub trzema gotowymi zawartościami ekranu. Program maszynowy (list. 1) liczy zaledwie 24 bajty, wymaga jednak jednego lub dwóch pomocniczych obszarów pamięci, tzw. buforów, każdy długości 6912 bajtów. W większości programów w języ-

```
C940    10 BUFOR2 EQU   51520
E440    20 BUFOR1 EQU   BUFOR2+6912
        30
FF40    40        ORG   BUFOR1+6912
        50
FF40    60 EKRAN1 LD    HL,BUFOR1
FF43    70        JR    EKRAN
        80
FF45    90 EKRAN2 LD    HL,BUFOR2
FF48   100 EKRAN  LD    DE,16384
FF4B   110 PETLA  LD    A,(DE)
FF4C   120        LD    C,(HL)
FF4D   130        LD    (HL),A
FF4E   140        LD    A,C
FF4F   150        LD    (DE),A
FF50   160        INC   HL
FF51   170        INC   DE
FF52   180        LD    A,D
FF53   190        CP    91
FF55   200        JR    NZ,PET
FF57   210        RET
```

```
 1 REM 3 EKRANY-PROGRAM LADUJACY
 2 REM ROLAND WACLAWEK  1985

10 CLEAR 51519: LET S=0
20 FOR A=65344 TO 65367
30 READ X: LET S=S+X: POKE A,X
40 NEXT A
50 IF S<>2091 THEN BEEP 1,1: STOP
60 DATA 33,64,228 24,3,33
70 DATA 64,201,17,0,64,26
80 DATA 78,119,121,18,35,19
90 DATA 122,254,91,32,244,201

100 REM PROGRAM DEMONSTRACYJNY
110 LIST 100: RANDOMIZE USR 65344
120 CLS : PRINT FLASH 1;"EKRAN 2"
130 RANDOMIZE USR 65349: CLS
140 PRINT AT 10,9;"EKRAN NUMER 3"
150 RANDOMIZE USR 65349: PAUSE 90
160 RANDOMIZE USR 65344: PAUSE 90
170 GO TO 150
```

ku BASIC, pracujących na mikrokomputerze ZX Spectrum 48KB lub Spectrum Plus, wygospodarowanie takiego bloku pamięci na ogół nie przysparza kłopotów.

Zasada działania programu "3 ekrany" polega na wymianie aktualnej zawartości pamięci ekranu z zawartością odpowiedniego bufora (rozmiar bufora odpowiada ściśle objętości pamięci ekranu wraz z obszarem atrybutów). Wymiana odbywa się bardzo szybko. Procedura maszynowa może być uruchamiana (wywoływana od adresów 65344 i 65349). Korzystając z pierwszego adresu, np. przez RANDOMIZE USR 65344, wymieniamy zawartość pamięci ekranu z pierwszym buforem, rozpoczynającym się od 58432. Uruchamiając procedurę od adresu 65349 powodujemy zamianę zawartości pamięci ekranu z treścią drugiego bufora, zlokalizowanego pod adresem 51520. Najłatwiej umieścić program maszynowy w pamięci za pomocą programu ładującego w języku BASIC (list. 2), do którego dołączono prosty przykład zastosowania. Jeśli błędnie wpisano linie DATA, program ładujący zamelduje o tym sygnałem akustycznym. Załadowany do pamięci program maszynowy "3 ekrany" można zapamiętać na taśmie magnetofonowej zleceniem SAVE "3 ekrany" CODE 65344, 24 i załadować ponownie przez LOAD "3 ekrany" CODE. Program jest przesuwny (relokowalny), może być więc uruchomiony także w innym obszarze pamięci.

Przed załadowaniem programu "3 ekrany" (z taśmy lub programem ładującym) należy zarezerwować odpowiedni obszar pamięci RAM, wyłączony z dyspozycji interpretera języka BASIC. Zamierzając wykorzystać obydwa bufory, wykonamy CLEAR 51519, zaś jeśli rezygnujemy z drugiego bufora, wykonamy CLEAR 58431. W tym drugim przypadku nie wolno korzystać z wywołania od adresu 65349, jednak dla programu i zmiennych języka BASIC pozostanie prawie o 7 KB pamięci więcej.

Program "3 ekrany" można wykorzystać, przygotowując na ekranie np. objaśniające teksty i rysunki i przechowując je w buforach. W razie potrzeby wywołujemy program maszynowy: na ekranie pojawiają się przygotowane wcześniej napisy lub grafika, a dotychczasowa treść ekranu umieszczona zostaje w buforze. Ponowne wywołanie programu maszynowego od tego samego adresu co poprzednio przywraca stan pierwotny. Jeśli w międzyczasie dokonaliśmy w sprowadzonym z bufora ekranie jakichś zmian, to zostanie on zapamiętany w zaktualizowanej postaci. Przygotowanie ekranów pomocniczych po rozpoczęciu pracy programu w języku BASIC nie zawsze jest wygodne. Ekrany graficzne lepiej tworzyć za pomocą odpowiednich programów narzędziowych. W takim wypadku zawartość ekranów pomocniczych można wprowadzić bezpośrednio do buforów np. z taśmy przez LOAD "Grafika" CODE 58432 ew. 51520. Pozwala to także uniknąć stopniowej budowy wprowadzanego z taśmy obrazu: blok grafiki wprowadzamy do bufora, a następnie, po skompletowaniu, przenosimy w ułamku sekundy na ekran wywołując program maszynowy.

Przy pierwszym wywołaniu programu "3 ekrany" może nastąpić wypełnienie ekranu przypadkową treścią (na ogół ściemnienie, jak przy włączeniu). Jest to normalne: wpisywana do pamięci ekranu początkowa zawartość bufora jest przypadkowa (na ogół same bajty o wartości 0).

ROLAND WACŁAWEK

PĘTLICZEK – bo pętla jest podstawą programowania.
Tu znajdziesz kolejną porcję zadań naszego Klubu Mistrzów Komputera.*)
MĘTLICZEK – bo znajdziesz tu różne różności, związane z minikomputerem tak cienką nitką, że Redakcja już nie bierze za nią odpowiedzialności.

---

*) regulamin KMK w numerze 2'86 naszego pisma.

[1] HISTORIA KOMPUTERÓW

Pierwsze były ręce

Ludzie liczyli od pradziejów. Już prehistoryczny pitekantrop liczył na to, że mamut wpadnie w pułapkę, a australopitek, że nie spadnie deszcz. Nasi praprzodkowie liczyli też swoje żony i dzieci.

Historię powstania komputerów zaczniemy więc w mrokach jaskiń. Najnowsze badania wskazują, że pierwszym urządzeniem do liczenia, a więc pierwszą maszyną matematyczną, była ręka liczącego. Było to przenośne urządzenie złożone z pięciu jednobitowych rejestrów, zwanych palcami. W tamtych czasach narzędzia nie były specjalizowane, a więc i ręce wykorzystywano również do wielu innych czynności.

Sposób liczenia był niezmiernie prosty. Liczenie wykonywano na otwartej dłoni zwróconej wnętrzem
w stronę oczu. W trakcie liczenia zaginano kolejne palce od prawej strony. W ten sposób liczbę 3 reprezentowano układem:

WWZZZ

gdzie W oznacza palec wyprostowany, a Z zgięty. Jak więc widać, maszyna pierwotnych zdolna była operować tylko liczbami całkowitymi w zakresie od 0 do 5. Widać tu wyraźny brak badań podstawowych. Dziś wiemy bowiem, znając dwójkowy układ liczenia, że 5 jednobitowych rejestrów starczy do przechowania liczb od 0 do $2^5 - 1 = 31$.

(cdn.)

KOMENTARZ? KOMENTARZ!

Spójrzcie na wydrukowany niżej program:

```
  10 DATA "@££ptt6TnT\\^???^\HH@
@"
  20 POKE 23675,PEEK (23635)+6:
POKE 23676,PEEK (23636)
  30 PRINT AT 10,15; INK 4;CHR$
144;AT 11,15; INK 2;CHR$ 145;AT
12,15; INK 2;CHR$ 146
  40 OVER 1: PLOT 121,72: DRAW 0
,18: OVER 0
  50 PLOT 120,95: PLOT 120,94: P
LOT 120,93
  60 PRINT AT 8,9;"JAKIES BURAKI
"
  70 PLOT 116,111: PLOT 116,112
  80 STOP
```

Na pierwszy rzut oka chyba nikt nie zgadł, co on robi, choć ma tylko 8 linii. Gdy bliżej przyjrzeć się temu programowi, też nic ciekawego nie widać. Po prostu brak dodatkowych informacji, czyli brak komentarza.

Można zaryzykować stwierdzenie, że dobry program to taki, który działa... Bardzo dobry program to dobry program, który rozwiązuje zadany problem, natomiast doskonały program to bardzo dobry program z dobrymi komentarzami.

Jako ćwiczenie w dobieraniu komentarzy proponuję rozgryźć i skomentować zamieszczony tu program. Najcelniejsze i właściwie położone komentarze, które zechcecie przysłać do nas, wydrukujemy.

Skoro już mowa o komentarzach, to warto zwrócić uwagę, że twórcy Logo, języka programowania mającego uczyć dobrych nawyków, zapomnieli o instrukcji komentarza. Aby móc komentować swoje dzieło w Logo, trzeba zaraz po wczytaniu kompilatora do komputera zdefiniować procedurę:

```
TO KOMENTARZ :lista
END
```

Teraz już w dowolnym miejscu definicji nowej procedury można umieścić:

```
KOMENTARZ [to jest właśnie komentarz]
```

i wszystko będzie jasne.

Przyzwyczajeni do komentarzy w Pascalu, którzy właśnie "przesiedli" się na Logo, powinni zdefiniować dwie procedury:

```
TO { : lista
END
```

oraz:

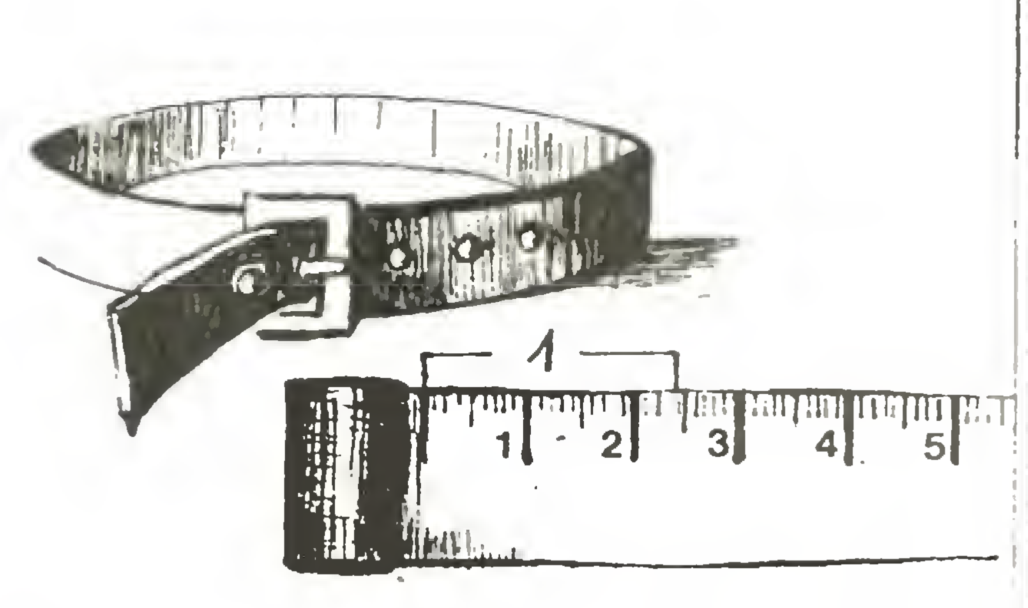
```
TO }
END
```

i mogą komentować:

```
{ [jak w Pascalu] }
```

PRAWA MURPHY'EGO W INFORMATYCE (i nie tylko)

PRAWO 1: Jeżeli tylko coś może się popsuć, to na pewno się popsuje.





ZADANIA KLUBOWE SERIA I, ZESTAW 3

1. Układankę Pentomino znamy chyba wszyscy. Jest to zestaw 12 płaskich klocków, zawierający wszystkie możliwe połączenia pięciu kwadracików jednostkowych. Taki zestaw klocków ma zadziwiającą własność. Wszystkie klocki można ułożyć w prostokąt 6x10 lub 5x12, lub 4x15, lub wreszcie 3x20 jednostkowych kwadracików.

Proponuję ułożyć algorytm lub napisać program odgadujący sposób ułożenia klocków Pentomina w którymś z podanych prostokątów. Oczywiście oczekujemy także rozwiązań uniwersalnych.

2. Jednym z ciekawych twierdzeń teorii liczb jest twierdzenie Lagrange'a o rozkładzie liczby naturalnej. Mówi ono, że każdą liczbę naturalną można przedstawić w postaci sumy czterech kwadratów. Na przykład:

$$5 = 2^2 + 1^2 + 0^2 + 0^2$$
$$300 = 16^2 + 6^2 + 2^2 + 2^2.$$

Proponuję ułożyć algorytm lub napisać program, który dla danej dodatniej liczby całkowitej znajduje jej rozkład na sumę czterech kwadratów.

3. Pisząc program w Basicu nie musimy deklarować zmiennych prostych. Jest to ułatwienie dla programisty, ale jednocześnie powoduje szybkie powstanie bałaganu. Często zdarza mi się przeglądać wszystkie linie programu w poszukiwaniu jakiejś zmiennej.

Proponuję więc napisać program działający na tekście programu w Basicu i wyszukujący zmienne. Program ten powinien wypisać wszystkie zmienne występujące w badanym programie oraz podać numery linii, w których one występują.

LESZEK RUDAK

5♦

import nowoczesnych podzespołów jest niezbędny, ponieważ przemysł krajowy nie może dostarczać jeszcze odpowiednich elementów. Do 1990 r. eksport będzie nadal wzrastał, lecz jego udział w całości produkcji będzie malał. Potrzeby kraju miały istotne znaczenie przy negocjowaniu zamówień rządowych na sprzęt komputerowy.

– *Kogo Urząd Postępu Naukowo-Technicznego i Wdrożeń wesprze finansowo? Czy tylko konkretnych producentów urządzeń elektronicznych, czy również tych, którzy powinni stosować i wdrażać informatykę do praktyki życia codziennego, np. służbę zdrowia, szkoły, poczty, banki?*

– Duże fundusze są przewidziane na zastosowania informatyki w bardzo różnych centralnych i resortowych programach badań, zarówno podstawowych jak i rozwojowych. W ofertach wstępnych oraz planach jest co najmniej kilkaset konkretnych tzw. celów realizacyjnych związanych z zastosowaniami informatyki. Każda mądra i efektywna propozycja będzie przez nas przyjęta. Z drugiej jednak strony wiele zgłoszonych tematów zostało odrzuconych lub będzie modyfikowanych. Te zespoły, które nadal mają ochotę pracować powoli i drogo, przeżyją bolesne rozczarowania.

– *Kto podejmuje decyzje o finansowaniu konkretnych prac nad wdrożeniami informatyki?*

– Proces podejmowania decyzji jest wieloszczeblowy i propozycje są analizowane zarówno przez specjalistów w Zespole Elektroniki, jak i przez członków Komitetu Nauki i Postępu Technicznego przy Radzie Ministrów. Korzystamy z pomocy ekspertów z uczelni, instytutów resortowych i placówek PAN. Wiele zależy od moich współpracowników z Zespołu Elektroniki. Nadal poszukujemy specjalistów do pracy w Urzędzie Postępu Naukowo-Technicznego i Wdrożeń.

– *W Polsce stworzone zostały doskonałe warunki (prawne i finansowe) dla rozwoju firm prywatnych i zagranicznych. Wiele przedsiębiorstw polonijnych produkuje różną aparaturę elektroniczną, w tym komputery...*

– Nie wiem, czy stworzone warunki można uznać za doskonałe, ale rzeczywiście dla przedsiębiorstw państwowych dostęp do dewiz uzyskanych przez nie z eksportu jest obecnie drogą cierniową. Taką sytuację wykorzystują przedsiębiorstwa polonijne, a państwowe – nie mogą działać równoprawnie. W ten sposób na rynek krajowy trafiają zupełnie przypadkowo dziesiątki typów mikrokomputerów, dla których nie istnieje serwis i dostęp do szerszego oprogramowania. Firmy polonijne, aby uzyskać wysokie dochody rzędu 4-8 tys. złotych za jednego dolara, sprowadzają tzw. kadłubowe konfiguracje, gdyż na dostawach urządzeń zewnętrznych zysk jest znacznie mniejszy.

Oczywiście, że firmy prywatne i zagraniczne działają dla własnego zysku, kierując się przede wszystkim prywatnym interesem. Filantropii oczekujemy natomiast od przedsiębiorstw państwowych, żądając od nich rezygnacji z opłacalnego eksportu urządzeń zewnętrznych dla komputerów i przeznaczania całej produkcji na rynek krajowy.

Dziękuję za rozmowę.

Rozmawiała: ELŻBIETA BOBROWSKA

7♦

szybkiego (i niekontrolowanego) zebrania dużej liczby przygotowanych po amatorsku programów. Dopiero kiedy w wyniku badań i studiów powstanie koncepcja optymalnego stosowania komputerów w procesie nauczania i wynikające zeń kryteria oceny programów komputerowych, dopiero kiedy te kryteria zostaną rozpowszechnione, będzie można pomyśleć o szerokim zaangażowaniu nauczycieli do tworzenia programów na zasadzie konkursów.

PLACÓWKI RESORTU OŚWIATY NIE POWINNY SIĘ BEZPOŚREDNIO ANGAŻOWAĆ W PRODUKCJĘ BĄDŹ ORGANIZACJĘ PRODUKCJI PROGRAMÓW

Zbędnym wysiłkiem byłoby tworzenie resortowej wytwórni. Wymagałoby to wielkich nakładów i doprowadziło do powstania monopolistycznego monstrum. Wzorem innych krajów (np. Wielka Brytania) produkcję oprogramowania dydaktycznego lepiej powierzyć firmom zawodowym na zasadzie opłacalności, bez dodatkowych nakładów ze strony oświaty. Stawia to resort oświaty w pozycji potężnego klienta, który może stawiać wymagania, wybierając najlepszy produkt u najlepszego producenta. Wszelkie dotowanie może być stosowane tylko doraźnie, dopóki wystarczy specjalnych środków, a zawsze przynosi szkodliwe skutki przez naruszenie relacji wartości. Natomiast dla zapewnienia wysokiej jakości produkcji i sterowania w kierunku właściwych rozwiązań, placówki oświatowe powinny zorganizować sprawny system oceny programów, skuteczną informację o nich a także instruktaż.

ŚRODKI NA OPROGRAMOWANIE DYDAKTYCZNE

jakimi będzie dysponowała oświata, powinny być przeznaczone na:

– prowadzenie badań i studiów nad metodyką stosowania mikrokomputerów w nauczaniu i kryteriami jakości oprogramowania

– przegląd i analizy porównawcze istniejących programów

– stałą, krytyczną ocenę wszystkich pojawiających się programów, na prowadzenie biblioteki i katalogu oprogramowania

– stałą informację i instruktaż docierający do wszystkich zainteresowanych nauczycieli na temat dostępnych programów.

Jeżeli na zakup gotowych programów dla szkół znajdą się środki w centralnym budżecie ministerstwa, to warto wprowadzić system, jaki z powodzeniem był stosowany we Francji: szkoły mające komputery dostawały pewną liczbę punktów upoważniających do otrzymania programów według wyboru, opłaconych z budżetu centralnego.

JAN DUNIN-BORKOWSKI

HISTORIA

Jako osoba grająca w brydża postanowiłem pewnego razu napisać program na rozdawanie kart w języku LOGO. Miałem przy tym ambicję napisania go w sposób możliwie elegancki, trzymając się czegoś, co nieprecyzyjnie można nazwać "duchem języka". Uważam bowiem, że szczególnie przy pisaniu procedur rekurencyjnych używających list powinniśmy zapisywać je w sposób funkcyjny, używając w miarę możliwości jak najmniej zmiennych, które mogą powodować efekty uboczne (w jednej procedurze możemy niechcący zmienić wartość zmiennej używanej w innej procedurze) i zmuszają nas do pamiętania, jakich zmiennych już używaliśmy. Nie wspominam o sytuacjach, gdy dwie osoby piszą fragmenty tego samego programu i później skaczą sobie do oczu wołając "co zrobiłeś z moim X?!".

Nie jestem jednak w tym względzie fundamentalistą, toteż rozpocząłem od zdefiniowania wysokości kart i ich kolorów:

MAKE "karty [A K D W 10 9 8 7 6 5 4 3 2]
MAKE "kolory [piki kiery kara trefle]

Talie kart najłatwiej reprezentować jako listę opisów kart w postaci par (wysokość kolor). Pisanie całej talii byłoby jednak zbyt kłopotliwe (jak wiadomo, lenistwo jest motorem postępu), przerzuciłem więc tę pracę na komputer.

Aby stworzyć talię, należy sporządzić listę złożoną z wszystkich możliwych par o pierwszym elemencie z listy "karty", a drugim z listy "kolory". Dla zachowania układu z oryginalnej, jeszcze nie rozpakowanej talii, gdzie karty ułożone są kolorami, dobierałem wartości kart do kolorów, a nie odwrotnie (LISTING 1).
Po napisaniu:
MAKE "talia TWORZ...TALIE
miałem już utworzoną talię. Należało ją teraz potasować, rozdać i wypisać ręce poszczególnych graczy. Procedurę ROZDANIE można zapisać na dwa sposoby (LISTING 2).

Oba te sposoby dałyby taki sam efekt i procedury pomocnicze byłyby dla nich identyczne. Mnie osobiście, jak zaznaczyłem na wstępie, bardziej podoba się drugi sposób. Zabrałem się następnie za tasowanie kart (a dokładniej za pisanie procedury do tasowania), nie podejrzewając żadnych kłopotów. I tu właściwie zaczyna się cała historia.

Tasowanie można zrealizować na wiele możliwych sposobów. Jeden z nich, który zastosowałem na początku, polega na tworzeniu z listy kart nowej listy, złożonej z kolejno losowanych kart z listy pierwszej. Pojedyncze tasowanie można powtórzyć wielokrotnie, dodałem więc parametr określający ilość tasowań. Oto moja procedura (LISTING 3)
(należy usunąć wylosowaną już kartę) (LISTING 4)

Procedura USUN jest nawet nieco ogólniejsza, niż wymagała tego sytuacja – dopisałem sprawdzenie, czy nie próbujemy usunąć nie istniejącego elementu. Zostało jeszcze losowanie (LISTING 5)

Czuję się tu zobowiązany do wyjaśnienia użycia zmiennej "wylosowana" w procedurze TASUJ...RAZ. Nie mogłem oczywiście napisać:

PEWNEGO ROZDANIA

1

```
TO TWORZ_TALIE
OUTPUT TWORZ :kolory :kárty
END
TO TWORZ :kolory :karty
IF EMPTYP :kolory [output []]
OUTPUT SE JEDEN_KOLOR FIRST :kolory :karty TWORZ BF :kolory :kar
ty
END
TO JEDEN_KOLOR :kolor :karty
IF EMPTYP :karty [output []]
OUTPUT FPUT SE :kolor FIRST :karty JEDEN_KOLOR :kolor BF :karty
END
```

2

```
TO ROZDANIE
MAKE "potasowana_talia TASUJ :talia
MAKE "rozdane_karty ROZDAJ :potasowa_talia
WYPISZ :rozdane_karty
END
TO ROZDANIE
WYPISZ ROZDAJ TASUJ :talia
END
```

3

```
TO TASUJ :ile_razy :talia
IF :ile_razy = Ø [output :talia]
OUTPUT TASUJ_RAZ TASUJ :ile_razy - 1 :talia
END
TO TASUJ_RAZ :talia
IF EMPTYP :talia [output []]
MAKE "wylosowana LOSUJ :talia
OUTPUT FPUT :wylosowana TASUJ_RAZ USUN :wylosowana :talia
END
```

4

```
TO USUN :element :lista
IF EMPTYP :lista [print [Nie ma!] OUTPUT []]
IF :element = FIRST :lista [output bf :lista]
OUTPUT FPUT FIRST :lista USUN :element BF :lista
END
```

5

```
TO LOSUJ :talia
OUTPUT ITEM 1 + RANDOM COUNT :talia :talia
END
```

6

```
TO TASUJ_DALEJ :karta :talia
OUTPUT FPUT :karta TASUJ_RAZ USUN :karta :talia
END
```

7

```
TO TASUJ_RAZ :talia
OUTPUT WLOZ FIRST :talia RANDOM 52 BF :talia
END
TO WLOZ :karta :gdzie :talia
IF :gdzie = Ø [output fput :karta :talia]
OUTPUT FPUT FIRST :talia WLOZ :karta :gdzie - 1 BF :talia
END
```

8

```
TO TASUJ_RAZ :talia
OUTPUT ROZDZIEL_I_POLACZ :talia [] []
END
TO ROZDZIEL_I_POLACZ :talia :kupka1 :kupka2
IF EMPTYP :talia [output se :kupka1 :kupka2]
IF ( RANDOM 2 ) = Ø [output rozdziel_i_polacz bf :talia fput fir
st :talia :kupka1 :kupka2]
OUTPUT ROZDZIEL_I_POLACZ BF :talia :kupka1 FPUT FIRST :talia :ku
pka2
END
```

OUTPUT FPUT LOSUJ :talia TASUJ...RAZ USUN LOSUJ :talia
:talia
ponieważ LOSUJ zwraca za każdym razem inny wynik. Nie znaczy to, że nie istnieje rozwiązanie bez użycia zmiennej. Zamiast drugiej i trzeciej linii procedury TASUJ...RAZ można napisać: OUTPUT TASUJ ...DALEJ LOSUJ :talia :talia
i zdefiniować procedurę TASUJ...DALEJ: (LISTING 6)
Nawiasem mówiąc jest to często używany chwyt, stosowany po to, by skomplikowane wyrażenie występujące w kilku miejscach tej samej instrukcji liczyć tylko raz, podczas obliczania argumentu dla pomocniczej procedury.

Zadowolony ze swoich poczynań napisałem wreszcie SHOW TASUJ 10 :talia
i... otrzymałem komunikat "Not enough space to proceed" (za mało pamięci, aby kontynuować). Sprawdziłem procedurę dla mniejszej ilości kart – działała poprawnie. Przyczyną było więc rzeczywiście ograniczenie pamięciowe dla języka LOGO na ZX Spectrum.

„Gwardia umiera, ale się nie poddaje" – zgodnie z tą dewizą z czasów napoleońskich postanowiłem nie odstępując od swoich zasad oprzeć się na innym algorytmie tasowania. Całkiem rozsądne wydaje się następujące postępowanie: bierzemy kartę z wierzchu talii i wkładamy ją losowo pomiędzy pozostałe karty. Kontynuując to postępowanie odpowiednią liczbę razy otrzymamy potasowaną talię. Napisałem zatem drugą wersję tasowania (procedura TASUJ została bez zmian) (LISTING 7)

Zdając sobie sprawę, że ten sposób tasowania wymaga większej liczby przetasowań (za każdym razem zmieniamy położenie jednej karty) napisałem
SHOW TASUJ 100 :talia

Niestety, wynik był ten sam: "Not enough space to proceed in TASUJ". Oczywiście – w TASUJ utworzyła się kolejka 100 procedur TASUJ RAZ czekających na swoją kolej (tym razem SHOW TASUJ 10 :talia przechodziło, ale było to ewidentnie za mało). Od ręki zmodyfikowałem procedurę TASUJ (LISTING 8)
Tym razem SHOW TASUJ' 100 :talia dało dobry wynik, choć gdzieś około trzeciej minuty oczekiwania zaczynałem się niecierpliwić. Procedura najwyraźniej działała zbyt wolno. Została mi jeszcze stara maksyma "do trzech razy sztuka". Zmobilizowałem szare komórki i wpadłem na trzeci pomysł tasowania. Polega on na odwróceniu procedury naturalnego tasowania, tj. rozdzielania talii na dwie kupki i losowego mieszania ich ze sobą (spróbujcie zrealizować "klasyczne" tasowanie) – postanowiłem rozdzielać talię na dwie losowe kupki i łączyć je ze sobą "doklejając" jedną do drugiej. Losowość występuje tu nie przy łączeniu, a przy rozdzielaniu kart. Procedurę tę również powinniśmy zastosować kilkakrotnie (TASUJ znów zostaje bez zmian) (LISTING 9)
(RANDOM 2 przybiera wartość 0 lub 1)

Tym razem z pewnym niepokojem napisałem
SHOW TASUJ 10 :talia
aby z zadowoleniem obejrzeć po dość rozsądnym czasie potasowaną talię, na razie jeszcze w postaci listy. Dalszy ciąg tej historii nie jest już tak pouczający – bez specjalnych kłopotów napisałem rozdawanie i wypisywanie kart (naturalnie według kolorów, uporządkowane wartościami). Nie będę tu podawał mojego rozwiązania, spróbujcie napisać swoje własne. Z całej tej opowieści wynika kilka morałów:

● Morał 1: nie należy się zniechęcać, gdy coś nam nie wychodzi.

● Morał 2: warto mieć zawsze na podorędziu odpowiednie maksymy, dopasowane do sytuacji.

● Morał 3 (na poważnie): dla każdego komputera istnieje zadanie, dla którego komputer ten okaże się niewystarczający. W praktyce jednak z rzeczywistymi ograniczeniami spotykamy się dość rzadko; najczęściej inne sformułowanie algorytmu pozwoli na rozwiązanie postawionego zadania. Nie musimy przy tym stosować rozwiązań trikowych (wbrew pozorom ilość znanych trików nie jest wyznacznikiem umiejętności pisania programów) ani rezygnować z czytelnej i eleganckiej postaci zapisanego algorytmu.

JAROSŁAW KANIA

O KOMPUTERACH I ∞ [3]

Szanowny Panie Redaktorze!

Dziękuję za odpowiedź na mój poprzedni list. Twierdzi Pan, że to, o czym pisałem, jest dla Pana tylko abstrakcyjnym wyobrażeniem matematyków o maszynach liczących i niewiele ma wspólnego z komputerami. Przeciwstawia Pan realny komputer, jako fizycznie istniejący i działający przedmiot, teoriom matematycznym istniejącym li tylko na papierze lub w głowach niektórych ludzi. Jedyny bliższy związek dostrzega Pan w nazwach.

Myślę więc, że przyszła pora na pokazanie, że ostatnio opisane twory matematyczne istnieją w realnych, jak Pan to nazywa, komputerach. Zanim jednak wskażę miejsce, gdzie widać automat w każdej maszynie (od ZX Spectrum do wielkich IBM-ów), pozwolę sobie na krótki wstęp. Otóż przypomnę pojęcie automatu, struktury, którą opisywałem w poprzednim liście. Dla matematyka automat to pewna struktura złożona z kilku powiązanych ze sobą zbiorów i relacji. Automat działa na ciągach znaków – przegląda je i decyduje czy badany ciąg ma pewne własności, czy też nie. Jeżeli ciąg posiada oczekiwaną własność, to automat zatrzymuje się w jednym ze swoich stanów końcowych i wtedy mówimy, iż akceptuje on dany ciąg. Jeżeli natomiast zatrzyma się w stanie, który nie jest końcowy, to powiemy, że automat odrzuca ciąg podany na wejściu. Automatem więc będzie każde urządzenie analizujące ciągi znaków i wydające werdykt: odrzucić lub zaakceptować.

Czas już teraz na wskazanie automatu o zdolności do akceptacji pewnych ciągów liter w komputerze, który stoi zapewne na Pana biurku. Załóżmy, że jest to zwykły Commodore 64, wyposażony w BASIC. Powiedzmy, że wpisał Pan kilka linii programu i wydał komendę RUN. Pierwsze, co teraz uczyni interpreter Basica Pańskiego komputera, to kontrola poprawności syntaktycznej wpisanego programu, czyli po prostu sprawdzenie, czy wpisany tekst jest rzeczywiście dopuszczalnym tekstem programu. Jeżeli interpreter stwierdzi, że tekst źródłowy jest poprawny, to przystąpi do wykonania podanych instrukcji. Jeśli natomiast tekst nie będzie podobny do żadnej z oczekiwanych postaci, to zasygnalizuje błąd i będzie czekać na dalsze komendy. W tym momencie interpreter zachowuje się jak opisany wcześniej automat akceptujący dany ciąg znaków. Jeżeli przyjrzymy się bliżej działaniu interpretera w czasie kontroli poprawności syntaktycznej, to dostrzeżemy, że jest to rzeczywiście automat. Rozpoznaje on litera po literze wpisany tekst. W zależności od tego, jaka jest zawartość rejestrów procesora (łącznie z licznikiem rozkazów) i jaką literę przeczytał, zmienia zawartość rejestrów i czyta następną literę. Gdy przeczyta całą linię, wydaje werdykt: akceptuje i przystępuje do realizacji lub odrzuca i sygnalizuje błąd. Dzięki temu instrukcja PRINT zostanie wykonana, natomiast PRENT – nie.

Oczywiście kontrolę poprawności syntaktycznej przeprowadza nie tylko interpreter Basica w komputerze Commodore 64. Wszystkie interpretery i kompilatory sprawdzają, czy podany tekst jest poprawnym tekstem w danym języku, a więc wszędzie tam występuje automat już jako twór realny. Można powiedzieć nawet więcej. Z pewnością Pan sam wielokrotnie wmontowywał automat do swoich programów. Przecież w najprostszej grze, w której sterujemy klawiszami: "A" – lewo, "L" – prawo, fragment programu badający, czy jednoliterowy ciąg wprowadzony z klawiatury składa się z jednej z tych liter, jest właśnie automatem.

Automaty takie, jakie opisałem w poprzednim liście (o jakich mówimy teraz), występują w wielu innych miejscach w programach komputerowych. Właściwie każde działanie komputera można przedstawić jako działanie pewnego automatu. Pojęcie automatu jest nawet ogólniejsze. Automaty akceptujące, podobne do opisanych, były wykorzystane przy próbie opisu sieci nerwowej. Dokonali tego w 1943 roku McCulloch i Pitts. Ich idea jest bardzo prosta. Rozważali system nerwowy jako zbiór neuronów wewnętrznych i neuronów czuciowych. W każdej elementarnej, niepodzielnej chwili czasu każdy neuron może być aktywny lub odpoczywać. Aktywny neuron wysyła impulsy do kilku innych neuronów. Mogą to być impulsy dwojakiego rodzaju: aktywujące lub dezaktywujące. Jeżeli w pewnej chwili dany neuron wewnętrzny otrzyma odpowiednią liczbę impulsów aktywujących, a nie otrzyma impulsu dezaktywującego, to w następnej chwili będzie aktywny. W przeciwnym przypadku w następnej chwili neuron będzie odpoczywał. Neurony czuciowe w stan aktywny może wprowadzić tylko bodziec zewnętrzny.

Załóżmy, że jakieś wydarzenie zewnętrzne, na przykład mucha chodząca po ręce, pobudza pewne neurony czuciowe. Po kilku chwilach naszego dyskretnego czasu pewne neurony wewnętrzne przejdą w stan aktywności. Jeżeli będą to neurony w naszym mózgu, odpowiedzialne za zmysł dotyku, to stwierdzimy: "mucha chodzi po ręce", czyli przyjmiemy pewną wiadomość, a więc zaakceptujemy pewien ciąg danych.

Taki model sieci nerwowej jest właśnie akceptorem, czyli automatem zdolnym do akceptowania pewnych ciągów danych. Zbiór stanów tego automatu to zbiór wszystkich możliwych konfiguracji aktywnych i odpoczywających neuronów czuciowych. Funkcję przejścia określają kierunki i rodzaje wysyłanych impulsów przez neurony aktywne.

Przedstawione tu badania McCullocha i Pittsa, i jeszcze wcześniejsze Shannona, leżały u podstaw teorii automatów. Drugim kierunkiem badań, który w istotny sposób wpłynął na powstanie tej teorii, było badanie efektywności w matematyce. Dla matematyka najważniejsze w każdej teorii są twierdzenia, które można w niej udowodnić. W teorii automatów jest kilka twierdzeń, odkrytych na samym początku jej powstania, które do dziś uchodzą za podstawowe. Twórcą jednego z nich jest Kleene. Twierdzenie to, opublikowane w 1956 roku, charakteryzuje wszystkie języki, które mogą być akceptowane przez automaty skończone. Muszę tu wyjaśnić, że język to po prostu zbiór słów, natomiast słowo – to skończony ciąg liter. Oczywiście mam na myśli abstrakcyjne litery np. zbiorem liter może być kolekcja znaczków pocztowych lub księgozbiór pewnej biblioteki. Język akceptowany przez automat to zbiór słów, które są akceptowane przez ten automat.

Nie będę wdawał się w szczegóły twierdzenia Kleene. Poprawne sformułowanie jest bowiem skomplikowane i prawdę mówiąc zrozumiałe tylko dla matematyków z kilkuletnim stażem. Języki akceptowane przez automaty nazywamy regularnymi. Takie języki dają się skonstruować ze słów jednoliterowych przy pomocy kilku operacji matematycznych na ciągach (suma zbiorów, uzupełnienia zbiorów, konkatenacja ciągów itp.). Najprostszym przykładem takiego języka jest język złożony ze wszystkich słów jednoliterowych. Językiem regularnym jest również język złożony ze wszystkich słów, a także każdy język skończony (tzn. skończony zbiór słów). Nie jest natomiast językiem regularnym np. język zawierający wszystkie słowa w danym alfabecie, których długość jest pełnym kwadratem liczby naturalnej.

Myślę, że to, co dzisiaj Panu przedstawiłem, przekona Pana, iż rzeczywistość matematyczna nie odbiega zbyt daleko od rzeczywistości komputerowej. Czysto matematyczne, teoretyczne rozważania są oczywiście bardzo abstrakcyjne, ale – jak widać – można za ich pomocą dość dobrze opisać realne przedmioty.

Pozdrawiam Pana i Czytelników Pańskiego pisma

MATEMATYK

Ps. Jak napisałem wyżej, każdy język skończony jest językiem regularnym, a więc nasz język ojczysty, składający się w tej chwili tylko ze skończonej liczby słów, jest językiem regularnym. Istnieje więc automat, który akceptuje tylko i wyłącznie słowa języka polskiego. Niestety, opisanie takiego automatu jest bardzo, bardzo kłopotliwe.

GIEŁDA

Przed aktualnym cennikiem kilka uwag o giełdzie sprzętu mikrokomputerowego odbywającej się w każdą pierwszą i trzecią niedzielę miesiąca w studenckim klubie STODOŁA w Warszawie.

Na początek bramka i bilet za 50 zł (czasem wchodzi się bez biletu?!). Potem ciasno, duszno i CIEMNO (normalnie jest tu kawiarnia z parasolami nad stolikami). Oferujący sprzęt i oprogramowanie siedzą na parapetach okien, przypadkowo ustawionych stolikach i krzesłach. Entuzjaści mikrokomputerów z rumieńcami na twarzach (od duchoty) przeciskają się między stolikami, nogami siedzących i ekspresem do kawy. Komputery grają (głośno), telewizory świecą i migają, w efekcie końcowym – koszmar, a komputer jest podobno zwiastunem XXI wieku.

Drogi Zarządzie klubu STODOŁA stańcie na poziomie sprzętu, jaki przyjęliście pod swój dach. Dochód z imprezy starczy na pewno na wynajęcie i posprzątanie większej i widniejszej sali.

Teraz cennik z dnia 4 maja 86 r.

- AMSTRAD 464 + zielony monitor — 240 tys. zł
  664 + zielony monitor — 360 tys. zł
  6128 brak ofert
- ATARI 600 XL — 80 tys. zł
  800 XL — 95 tys. zł
  130 XE — 160 tys. zł
  magnetofon (trudny do kupienia) — 25-30 tys. zł
  (coraz więcej sprzętu firmy Atari na rynku za coraz mniejsze pieniądze)
- COMMODORE C128 — 320 tys. zł
  C64 — 175 tys. zł
  stacja dysków 1541 — 150-170 tys. zł
  drukarka MPS 803 — 140 tys. zł
  VIC 20 + magnetofon — 50 tys. zł
  SHARP MZ 700 bez drukarki — 180 tys. zł
- ZX Spectrum 128K w wersji angielskiej — 245 tys. zł
- ZX Spectrum + — 110 tys. zł
- TIMEX 1000 (ZX 81) — 25 tys. zł
- drukarka SEIKOSHA GP 50 A do Spectrum — 120 tys. zł
  (coraz mniej komputerów Spectrum na rynku)
- Do Spectrum
  Interfejs 2 — 24 tys. zł
  Kempston interfejs — 9-15 tys. zł
  Kempston joystick — 6-9 tys. zł
- Interfejs i joystick na podczerwień do Spectrum — 24 tys. zł
- Oprogramowanie
  Turbo coopy do Atari — 5-7,5 tys. zł
- Dyskietki 5,25 cala — 1,2-1,7 tys. zł
- Dyskietki 3 cale do Amstrada — od 2,5 tys. zł

Nowość na rynku to literatura do IBM PC po 15 zł za stronę oraz oprogramowanie użytkowe i gry. Ceny (wysokie) do uzgodnienia z bezpośrednio zainteresowanymi.

(Informacje zebrał i uwagami się podzielił: ZR)

Tak natomiast przedstawiały się ceny z 23 maja w łódzkim "Domarze":

ZX 81 16 KB	35 000 zł
ZX Spectrum 48K	120 000 zł
ZX Spectrum Plus	175 000 zł
Atari 600 XL	80 000 zł
Atari 800 XL	100 000 zł
Atari 130 XE	180 000 zł
Commodore 116	90 000 zł
Commodore 64	190 000 zł
Commodore 64 z magnetofonem	230 000 zł
Commodore 128	400 000 zł
Commodore 128 D	750 000 zł
Amstrad / Schneider 464	360 000 zł
Amstrad / Schneider 646	650 000 zł
Amstrad / Schneider 6128 G	800 000 zł
Amstrad / Schneider 6128 C	900 000 zł
Amstrad / Schneider 8256	1 200 000 zł

TAJWAŃSKIE SMOKI NA POLSKIM RYNKU

Firmy wysyłkowe z Tajwanu zdobyły na naszym rynku pozycję dominującego dostawcy systemów zgodnych z IBM PC: od nich pochodzi większość sprzętu oferowanego przez firmy polonijne, rzemieślnicze i Dom Handlowy Nauki. Według niektórych szacunków do Polski trafiło już z Dalekiego Wschodu ponad 5 tys. IBM-podobnych systemów. Chociaż w handlu tym uczestniczy kilkanaście firm tajwańskich, to jednak dwie z nich zdobyły sobie pozycję dominującą. Jedna, oznaczona tu przez (I), proponuje wyższe ceny. Wkroczyła ona jednak na nasz rynek jako pierwsza i ma już wyrobioną opinię solidnego dostawcy. Druga, oznaczona tu przez (D), oferuje ceny niewiarygodnie niskie i bardzo ofensywnie stara się dotrzeć do polskich klientów. Proponowane przez nią ceny są tak niskie, że stanowią realną konkurencję dla sprowadzanych z Europy Zach. komputerków 8-bitowych: Atari, Amstrada i Commodore, zwłaszcza gdy w grę wchodzi system do prac profesjonalnych, z odpowiednią pamięcią i wyposażeniem.

Koszty i warunki przesyłki są następujące: producent gwarantuje dostawę w ciągu 15 dni (I), 10 dni (D) od otrzymania płatności. W praktyce terminy te są dłuższe – do 6 tygodni – ze względu na czas transferu pieniędzy oraz oczekiwanie w kolejce na samolot. Za transport i ubezpieczenie dostawca (I) dolicza: 10 dol. za kg (przy paczkach do 45 kg) lub 3.3 dol. za kg (przy paczkach ponad 45 kg). Waga typowego zestawu: 15 kg komputer, 9 kg monitor. Dwa zestawy wystarczają do przekroczenia granicy 45 kg.

Drugi dostawca (D) dolicza 80 dol. do każdego komputera w wypadku partii mniejszych niż 4 sztuki. Dla większych partii cena obejmuje koszty dostawy. W praktyce do tej ceny należy doliczyć kilkanaście tysięcy złotych płatne na lotnisku w Warszawie za różnego rodzaju nieokreślone usługi polskich spedytorów.

(WM)

Lp.	SPRZĘT	(D)	(I)
1.	IBM PC XT 640 KB, 2 napędy 5,25 cala, karta wielofunkcyjna, karta monochromatyczna	445	660
2.	IBM PC XT portable, 640 KB, karta Herkules, zielony monitor 5 cali	550	-
3.	IBM PC AT 512 KB na głównej karcie + karta 3 MB, napęd 1,2 MB, twardy dysk 20 MB	1700	2600

Lp.	POSZCZEGÓLNE ELEMENTY		(D)	(I)
1.	Obudowa:	XT AT	16 -	25 56
2.	Zasilacz:	135 W 200 W	36 80	49 88
3.	Klawiatura:	XT AT	32 45	42 50
4.	Główna płyta z 640 KB, 8 złączy		70 (bez kości pamięci)	200-245 (w zależności od częstotliwości zegara CPU)
5.	AT główna płyta, 6 MHz, 640 KB		550	620
6.	Napęd dyskowy:	360 KB 720 KB 1,2 MB (AT)	60 - 120	- 100 140
7.	Karta:	wejścia/wyjścia graficzna Herkules kontrolera Winchester	50 45 (monochrome) 130	75 70 170
8.	Twardy dysk 20 MB		340	860 (40 ms) 580 (85 ms)
9.	Proste karty jednofunkcyjne (RS-232, joystick, zegar, printer)		10	15-19
10.	AT karta wielofunkcyjna 3 MB (bez kości)		130	150
11.	Monitor:	zielony RGB	57 240	70 260